2003

Wavelet-based simulation of geological variables

T. T. Tran
Edith Cowan University

Wavelet-based Simulation Of Geological Variables

T. T. Tran
Edith Cowan University

Edith Cowan University

Copyright Warning

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

WAVELET-BASED SIMULATION

OF

GEOLOGICAL VARIABLES

Doctor of Philosophy (Mathematics)

T. T. Tran

FACULTY OF COMPUTING, HEALTH AND SCIENCE

EDITH COWAN UNIVERSITY

WESTERN AUSTRALIA

November 2003

ABSTRACT

This thesis introduces a number of conditional simulation algorithms using wavelet bases. These make use of two orthogonal wavelet bases, the Haar and the Db2 bases. Firstly, two single-level algorithms are introduced, HSIM with the Haar basis and DB2SIM with the Db2 basis. HSIM reproduces the histogram and semivariogram model of isotropic samples but not the semivariogram model of anisotropic samples. DB2SIM reproduces the histogram and semivariogram model in both the isotropic and anisotropic cases but, because of the conditioning method employed, is not as computationally efficient as we would wish.

Because of the limitations of HSIM and Db2SIM two multi-level wavelet-based conditional simulation algorithms PWSIM and DWSIM have then been developed. In PWSIM, the conditional realisations are obtained by post-processing non-conditional realisations generated via an available non-conditional simulation algorithm using kriging. In DWSIM the data are conditioned directly via properties of the discrete wavelet transform. Because of the conditioning method, DWSIM is faster than PWSIM.

The performance of PWIM and DWSIM with respect to the Haar and the Db2 wavelet bases is assessed via the local and global accuracy of the results. Both quantitative and visual assessments indicate that, for both wavelet bases, the realisations obtained via PWSIM have more variability than those obtained via DWSIM. If the Haar basis is used, PWSIM and DWSIM perform equally well. If the Db2 basis is used then PWSIM performance is much better than DWSIM performance.

For both PWSIM and DWSIM, the use of the Db2 basis rather than the Haar basis increases the computational effort without producing a comparable increase in

algorithm performance. In PWSIM the use of the Db2 basis slightly improves algorithm performance but the use of the Db2 basis in DWSIM decreases algorithm performance.

A performance comparison between DWSIM using the Haar basis and the commonly used conditional simulation algorithm SGSIM shows that DWSIM produces results that are at least as good as those obtained by SGSIM but with less computational effort. The computational advantage of DWSIM over SGSIM is especially pronounced when a large number of realisations are simulated. In addition, the result obtained via DWSIM does not depend on user defined parameters as is the case in both SGSIM and PWSIM.

The final result here is a (Haar) wavelet-based conditional simulation algorithm DWSIM that performs well in both the isotropic and the anisotropic cases and, particularly when simulating a large number of realisations, is much faster than the standard algorithm in common use.

DECLARATION

I certify that this thesis does not, to the best of my knowledge and belief,

i. incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education,

ii. contain any material previously published or written by another person except where due reference is made in the text; or

iii. contain any defamatory material.

Signature:

Date: 12 November 2003

ACKNOWLEDGEMENTS

I owe a debt of gratitude to many people without whom the completion of this thesis would not have been possible.

First and foremost, I would like to express my sincere thanks to my principal supervisor Dr. Ute Mueller and my co-supervisor Associate Professor Lyn Bloom for their encouragement and for spending their precious time, energy and patience to proofread, criticise and correct this thesis. Special thanks are given to my principal supervisor who has unconditionally offered her endless patience guiding and supporting my work.

Next, I would like to thank all staff of the School of Engineering and Mathematics for their prompt help during my research.

Finally, I greatly appreciate my husband and my daughter who have understood the work of a PhD student and for many years have frequently consumed burnt meals without complaint.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Geostatistical methods are now an essential tool in assessing the economic viability of a mineral deposit or an oil reservoir. The outcome of the geostatistical analysis of a geological variable, for example gold grade in an ore body, determines whether a deposit will be mined. It is thus important to continuously develop improved geostatistical methods that enable more accurate prediction of the attribute of interest.

We begin with the assumption that spatial data cannot be considered to be independent. Therefore, from a sample taken in a region the experimental spatial continuity is measured and then modelled. This model is used for the prediction of the unknown values in that region by estimation or simulation. Estimation methods are based on generalised linear regression methods and predict an "expected value" at each location. They are known as kriging techniques and produce smoothed images that, due to the nature of the input data, often under-estimate high grades and over-estimate low grades.

Because of the smoothing effect, kriging is not an appropriate technique in applications that focus on the extreme values. For example, in assessing the impact of lead in a polluted region on health using kriging, the health costs can be underestimated because of the smoothing effect. In such cases, simulation is an alternative approach since unlike kriging, simulation methods reproduce the statistical fluctuations of the phenomenon and the simulated images are not necessarily smooth.

Unlike kriging, from which the estimation is unique, a simulation algorithm generates a set of "equiprobable" realisations. Therefore, simulation is a useful tool in risk assessment or decision making. For example, in petroleum applications, geostatistical simulation can be used to compute the distribution curves at well locations. These probability distributions allow one to evaluate the range of values at any potential

well location, and hence decide whether or not to drill a new well.

The aim of geostatistical simulation algorithms is to model reality by generating equiprobable realisations that reproduce relevant properties of a given sample. Such properties include variance, histogram and spatial correlation. If the simulated values at sample locations are equal to the actual sample values then the simulation is said to be conditional; otherwise it is called nonconditional. Because the simulated values at sampled locations are equal to the actual data values, conditional simulation is of more interest than non-conditional simulation and can be applied in the modelling of ore deposits.

All of the conditional simulation algorithms developed to date, for example the LU decomposition simulation, simulated annealing, sequential Gaussian simulation (Chilès and Delfiner, 1999) and conditional spectral simulation (Yao, 1998) are subject to computational drawbacks. In the LU decomposition algorithm, simulated realisations are obtained by the decomposition of the associated covariance matrix. The limitation of the method is related to the size of the covariance matrix to be handled. When it was first developed, this algorithm could only simulate realisations with up to 700 points (Davis, 1987). Today, with the development of computer CPU, it can be used to simulate realisations of larger size. However, because of its costly computational effort this is still a time consuming algorithm.

The simulated annealing algorithm, from which simulated realisations are obtained by exploiting an optimisation method, is also a computationally expensive algorithm. The result of the simulation also depends on the annealing schedule defined the user. The annealing schedule includes the initial temperature, the reduction factor, which is used to reduce the temperature and the maximum number of perturbations at a given temperature. The rate of convergence depends on the reduction factor, the smaller

the reduction factor the faster the temperature is reduced and, hence, the faster the rate of convergence. The user has to define a suitable reduction factor for the cooling process since an inappropriate choice of a high reduction factor will unnecessarily slow down the rate of convergence and an inappropriate choice of a low reduction factor can lead to a risk that the result is trapped in unacceptable sub-optima (Goovaerts, 1997).

The results obtained by the most frequently used method, sequential Gaussian simulation (SGSIM), which will be detailed in Chapter 2, also depend on many parameters selected by the user. In SGSIM the simulation is carried out sequentially along a random path that visit each grid node in the study region exactly once. At each location on the random path the simulated value is drawn from a normal distribution with mean equal to the kriging estimate and variance equal to the kriging variance. The result of the simulation, therefore, depends on the kriging method selected by the user. In addition, the search radius or the number of original and previously simulated data used for simulating a value also impacts the outcome of the simulation. For example, the choice of too small a search radius may lead to a poor conditioning or an inappropriate reproduction of the spatial correlation. Therefore, to ensure a large enough search radius this algorithm requires that the size of the study region to be simulated be larger than the range of the semivariogram (Vann et al, 2001).

The conditional spectral simulation algorithm by Yao (1998) is fast since it is based on the fast Fourier transform. However, the conditioning of the realisations using Yao's algorithm introduces artefacts in the neighbourhood of conditioning data. High data values are surrounded by low simulated values and vice versa. In comparing the simulated values with jackknife data, the point-to-point correlation between the

simulated values and the true values is close to zero. This algorithm also uses an optimisation method in conditioning the data; hence, there is a risk that the iterative process does not converge if the spectral density is not consistent with the data.

Therefore, it is necessary to develop a conditional simulation algorithm that requires less computational effort and is not subject to the above disadvantages. This thesis introduces two conditional simulation algorithms, called PWSIM and DWSIM, one of which, the conditional simulation algorithm DWSIM, satisfies both of the above requirements. Both PWSIM and DWSIM make use of the properties of wavelets to generate realisations of a random function. However, the method of obtaining conditional simulated realisations in the two algorithms is different. In PWSIM the data are conditioned by a postprocessing step using kriging and in DWSIM the data are conditioned directly using the discrete wavelet transform. Due to this method DWSIM is a very effective conditional simulation algorithm.

Compactly supported wavelets are families of functions that take non-zero values only for a small value range. As in the case of Fourier analysis, there exists a fast algorithm called the discrete wavelet transform which can be used to save the computer storage space and computational time. The discrete wavelet transform computes the scaling and wavelet coefficients at a coarser scale from the scaling coefficients at the previous finer scale. It is initialised by regarding a set of evenly-spaced discrete values as scaling coefficients at the original scale. On each application of the discrete wavelet transform, the size of the set of scaling coefficients is compressed in one-dimensional space by a factor of two and in two-dimensional space by a factor of four.

Because the discrete wavelet transform is fast, a simulation algorithm that makes use of this transformation is expected to be a fast algorithm. In addition, in compressing the set of scaling coefficients, the discrete wavelet transform also compresses

the spatial correlation of these coefficients. Because of the localisation property of wavelets, the spatial correlation is limited only to a few scaling coefficients in a neighbourhood. Therefore, an algorithm based on the discrete wavelet transform does not require that the size of the simulated region be larger than the range of the semivariogram. The discrete wavelet transform is also independent of user-defined parameters and of the number of previously simulated values.

Because of the properties discussed above, wavelets are used as the basis of the conditional simulation algorithm DWSIM. This is a multi-level wavelet-based conditional simulation algorithm since it starts the simulation at some coarse scale then reconstructs the simulated realisations using the correlation among the scaling coefficients. Two orthogonal wavelet bases can be used in DWSIM, the Haar and the Db2 wavelet bases. Haar wavelets are discontinuous functions whose filters have only two non-zero coefficients. Db2 wavelets are continuous functions whose filters have four non-zero coefficients. Both wavelet bases require less computational effort compared with other families of orthogonal compactly supported wavelets.

The Haar wavelets are chosen as the basis of DWSIM since the computation of the discrete wavelet transform based on the Haar filters is not subject to the edge effect. However, because of the discontinuity of Haar wavelets, the Db2 basis is also introduced in this thesis for comparison purposes. The impact of the choice of wavelet basis used in DWSIM is investigated by comparing the results obtained from DWSIM using the Haar basis with those obtained by DWSIM using the Db2 basis. The results of the comparison show that DWSIM using the Haar basis is more efficient.

For comparison purposes, we implement a multi-level nonconditional simulation algorithm using wavelet analysis called WSIM introduced by Zeldin and Spanos (1995). The nonconditional realisations obtained using this algorithm are postpro-

cessed using kriging to obtain conditional simulations. The associated conditional simulation algorithm that combines WSIM and the postprocessing step is called PWSIM. The performance of PWSIM then is compared with the performance of DWSIM using different measures to evaluate the effectiveness of the two wavelet-based algorithms.

The thesis consists of 12 Chapters. Chapters 2 and 3 give the background on geostatistics and wavelet analysis. In Chapter 2, the concept of the random function model on which geostatistical methods are based is introduced. Next, the measures of the spatial continuity of sample data such as the experimental covariance, experimental correlation and experimental semivariogram are presented. This leads into the modelling of the spatial continuity based on the semivariogram model which is then used to compute the covariance model used in the simulation. Finally, the kriging and simulation methods used to deduce the values at unknown locations are discussed. Two types of kriging are introduced: simple kriging and ordinary kriging. Simple kriging is used in the sequential Gaussian simulation algorithm, also discussed in this Chapter, whose performance is compared with DWSIM performance in Chapter 10. Ordinary kriging is used in the postprocessing step to obtain conditional simulated realisations of PWSIM.

Chapter 3 includes deterministic and stochastic wavelet analysis in one- and two-dimensional spaces. The deterministic wavelet analysis consists of the construction of wavelets and the formulation of the discrete and inverse discrete wavelet transforms. The stochastic wavelet analysis consists of the computation of the covariances of pairs of scaling and wavelet coefficients. These covariances are used to simulate the unknown scaling and (or) wavelet coefficients from the known ones. In PWSIM these covariances are used to simulate the scaling coefficients at some coarser scale and

to compute the wavelet coefficients at the same scale from those scaling coefficients. The inverse discrete wavelet transform is used to reconstruct the scaling coefficients at the finer scales. In DWSIM the discrete wavelet transform is used to condition the data and the covariances of pairs of scaling coefficients are used to reconstruct scaling coefficients at a finer scale from those at the next coarser scale. In addition with the covariances, the wavelet spectrum and the scaling spectrum are also introduced. These spectra are applied in the single level simulation algorithms HSIM and DB2SIM introduced in Chapter 6.

The performance of the simulations algorithms discussed in this thesis is assessed using the measures presented in Chapter 4. These measures are both quantitative and visual. The quantitative measures are based on the accuracy, goodness and precision of the probability distribution of the simulated values, misclassification analysis, E-type estimates, conditional variances and the measures for the reproduction of the semivariogram model and the histogram of the associated sample. Visual measures are based on the mosaic plots of simulated realisations.

Chapter 5 provides the description of two data sets, one isotropic and one anisotropic, their normal transformation and samples taken from these. In the isotropic case, five samples (one gridded and four ungridded) drawn from the normal scores of the isotropic data set are used to test the performance of the algorithms in the case that the underlying random function is multivariate standard normal. Algorithm performance in practical applications (the data are not necessarily drawn from multivariate normal distribution) is tested using an ungridded sample drawn from the attribute values of the isotropic data set. In the anisotropic case, in the case that the underlying random function is multivariate standard normal, performance is tested using two samples, one gridded and the other is ungridded, drawn from the normal scores

of the anisotropic data set. In practical applications, algorithm performance is tested using an ungridded sample drawn from the attribute values of the anisotropic data set.

Chapter 6 introduces two single level wavelet-based algorithms used to generate conditional realisations and discusses the performances of these algorithms. Chapter 7 gives the description, applications and general performance assessment of the multi-level wavelet-based conditional simulation algorithms PWSIM and DWSIM. The algorithm description includes the formulation for the simulation step and the computation of the related weights. The applications of PWSIM and DWSIM using the Haar basis and the Db2 basis are carried out in both isotropic and anisotropic cases. In each case, algorithm performance is assessed visually via the mosaic plots and the plots of the experimental semivariograms and histograms of three typical realisations.

A detailed performance comparison between PWSIM and DWSIM using each of the wavelet bases in the case when the underlying random function is multivariate standard normal with regard to sampling method and data structure is discussed in Chapter 8. The impact of choice of wavelet basis and computational effort are also compared. The performance measures used in this Chapter are the accuracy, goodness and precision of the distributions of simulated values and the reproduction of the associated semivariogram models and sample histograms. The results of the comparison show that DWSIM using the Haar basis is an effective conditional simulation algorithm. Hence, we discard the algorithm PWSIM and focus only on the influence of the sample size on DWSIM performance using the Haar basis in Chapter 9.

Chapter 10 provides the performance comparison between DWSIM using the Haar basis and SGSIM in practical applications, that is, in the case where the underlying

random function is not multivariate standard normal. The performance measures are the reproduction of the semivariogram models and sample histograms, misclassification analysis, E-type estimates and conditional variances.

Chapter 11 provides the concluding discussion and Chapter 12 consists of the appendices which include the accompanying CD that contains the executable programs WSIM and DWSIM and the data sets.

2 GEOSTATISTICAL BACKGROUND

This Chapter introduces the theory of random functions in geostatistics which provides the framework for the modelling of spatial data. Unless otherwise stated, the background given in this Chapter is adopted from Goovaerts, 1997.

2.1 The Random Function Model

One goal of geostatistical methods is to model an attribute of interest at unsampled locations. Most geostatistical methods are based on the concept of the random function model. In geostatistics, the value of an attribute at location $\mathbf{u}$ in a study region $\mathcal{A} \subseteq \mathbb{R}^n$ is denoted by $z(\mathbf{u})$. Ore grade and soil type are examples of such attributes. In the random function model concept, the value $z(\mathbf{u})$ is regarded as a particular value of an unknown random variable $Z(\mathbf{u})$ at the point $\mathbf{u}$. A random function Z then is defined as the collection of all random variables in the study region

$$Z : \mathcal{A} \longrightarrow \{Z(\mathbf{u}) : \mathbf{u} \in \mathcal{A}\}.$$

A random function Z can be characterised by its first- and second-order moments. The first-order moment or expected value, if it exists, of a random variable $Z(\mathbf{u})$ is a function of $\mathbf{u}$ and is denoted by

$$E\{Z(\mathbf{u})\} = m(\mathbf{u}). \tag{1}$$

There are three second-order moments used in geostatistics: the variance, the covariance and the semivariogram.

1. The variance, if it exists, of a random variable $Z(\mathbf{u})$ is defined by

$$Var[Z(\mathbf{u})] = E\{[Z(\mathbf{u}) - m(\mathbf{u})]^2\}. \tag{2}$$

In general, the variance depends on the location $\mathbf{u}$. Note also that we can write

$$Var\{Z(\mathbf{u})\} = E\{Z(\mathbf{u})^2\} - m(\mathbf{u})^2. \tag{3}$$

2. The covariance of two random variables $Z(\mathbf{u})$ and $Z(\mathbf{u}')$ is a function of locations $\mathbf{u}$ and $\mathbf{u}'$ and is given by

$$C(\mathbf{u}, \mathbf{u}') = E\{[Z(\mathbf{u}) - m(\mathbf{u})][Z(\mathbf{u}') - m(\mathbf{u}')]\}. \tag{4}$$

3. The semivariogram of random variables $Z(\mathbf{u})$ and $Z(\mathbf{u}')$ is defined as

$$\gamma(\mathbf{u}, \mathbf{u}') = \frac{1}{2} Var[Z(\mathbf{u}) - Z(\mathbf{u}')] \tag{5}$$

and similar to the covariance, in general, the semivariogram depends on locations $\mathbf{u}$ and $\mathbf{u}'$.

A random function Z is said to be *strictly stationary* if the joint distribution of $\{Z(\mathbf{u}_1), ..., Z(\mathbf{u}_n)\}$ is the same as the joint distribution of $\{Z(\mathbf{u}_1 + \mathbf{h}), ..., Z(\mathbf{u}_n + \mathbf{h})\}$ for all $\mathbf{h}$. If the expected value of the random function is a constant and the covariance between two random variables $Z(\mathbf{u})$ and $Z(\mathbf{u} + \mathbf{h})$ is finite and depends only on the separation vector $\mathbf{h}$, that is

$$E\{Z(\mathbf{u})\} = m \tag{6}$$

and

$$C(\mathbf{h}) = E\{Z(\mathbf{u} + \mathbf{h})Z(\mathbf{u})\} - m^2 \tag{7}$$

for all $\mathbf{u} \in A$ then the random function Z is said to be *second-order stationary*. A random function Z is said to be *intrinsic* stationary if the increments $Z(\mathbf{u} + \mathbf{h}) - Z(\mathbf{u})$ are second-order stationary. A second-order stationary random function is also

intrinsic stationary but the converse is not true. The *semivariogram* $\gamma(\mathbf{h})$ of a second-order stationary random function is defined by

$$\gamma(\mathbf{h}) = \frac{1}{2}E\{(Z(\mathbf{u}+\mathbf{h}) - Z(\mathbf{u}))^2\}, \tag{8}$$

and it is related to the covariance $C(\mathbf{h})$ by

$$\gamma(\mathbf{h}) = C(0) - C(\mathbf{h}). \tag{9}$$

The standardised form of the covariance is called the *correlogram* $\rho(\mathbf{h})$ and is related to the semivariogram by

$$\rho(\mathbf{h}) = \frac{C(\mathbf{h})}{C(0)} = 1 - \frac{\gamma(\mathbf{h})}{C(0)}. \tag{10}$$

In this thesis we will assume that the underlying random function is second-order stationary. In other words, the correlation between two random variables $Z(\mathbf{u}+\mathbf{h})$ and $Z(\mathbf{u})$ depends only on the separation vector $\mathbf{h}$.

2.2 Sample Spatial Correlation

In geostatistics the analysis or the mapping of the spatial distribution of the values of an attribute z in a region A is based on a sample of n data $z(\mathbf{u}_\alpha), \alpha = 1, 2, ..., n$ taken from that region. The spatial continuity of the sample is measured and this measurement is then used for modelling the spatial continuity of the attribute over the whole region. The tools used to describe the spatial continuity of sample data include the *experimental covariance*, *experimental correlation* and *experimental semivariogram* functions.

The experimental covariance and the experimental correlation functions are measures of the similarity between data separated by a vector $\mathbf{h}$ called a *lag*. The exper-

imental covariance function $\widehat{C}(\mathbf{h})$ is defined by

$$\widehat{C}(\mathbf{h}) = \frac{1}{N(\mathbf{h})} \sum_{\alpha=1}^{N(\mathbf{h})} [z(\mathbf{u}_\alpha) z(\mathbf{u}_\alpha + \mathbf{h}) - \hat{m}_{-\mathbf{h}} \hat{m}_{+\mathbf{h}}], \tag{11}$$

where $N(\mathbf{h})$ is the number of data pairs of separated by $\mathbf{h}$; $\hat{m}_{-\mathbf{h}}$ and $\hat{m}_{+\mathbf{h}}$ respectively are the means of the tail $z(\mathbf{u}_\alpha)$ and the head $z(\mathbf{u}_\alpha + \mathbf{h})$ values corresponding to all pairs $(z(\mathbf{u}_\alpha), z(\mathbf{u}_\alpha + \mathbf{h}))$ and are defined by

$$\hat{m}_{-h} = \frac{1}{N(\mathbf{h})} \sum_{\alpha=1}^{N(h)} z(\mathbf{u}_\alpha)$$

and

$$\hat{m}_{+h} = \frac{1}{N(\mathbf{h})} \sum_{\alpha=1}^{N(h)} z(\mathbf{u}_\alpha + \mathbf{h}).$$

The experimental correlation function $\widehat{\rho}(\mathbf{h})$, taking values in the interval $[-1, 1]$, is the standardised form of the experimental covariance function given by

$$\widehat{\rho}(\mathbf{h}) = \frac{\widehat{C}(\mathbf{h})}{\widehat{\sigma}_{-\mathbf{h}} \widehat{\sigma}_{+\mathbf{h}}}, \tag{12}$$

where $\widehat{\sigma}_{-\mathbf{h}}$ and $\widehat{\sigma}_{+\mathbf{h}}$ are respectively the standard deviations of the tail and the head values. They are defined by

$$\widehat{\sigma}_{-\mathbf{h}} = \sqrt{\frac{1}{N(\mathbf{h})} \sum_{\alpha=1}^{N(h)} (z(\mathbf{u}_\alpha))^2 - \hat{m}_{-\mathbf{h}}^2}$$

and

$$\widehat{\sigma}_{+\mathbf{h}} = \sqrt{\frac{1}{N(\mathbf{h})} \sum_{\alpha=1}^{N(h)} (z(\mathbf{u}_\alpha + \mathbf{h}))^2 - \hat{m}_{+\mathbf{h}}^2}.$$

When the experimental covariance $\widehat{C}(\mathbf{h})$ is a function of both the magnitude $|\mathbf{h}|$ and the direction of the vector $\mathbf{h}$, the function is said to be *anisotropic*. If this function depends only on $|\mathbf{h}|$, then the random function is *isotropic*, that is, the spatial variability is identical in all directions of space.

The experimental semivariogram measures the average dissimilarity and is defined by

$$\hat{\gamma}(\mathbf{h}) = \frac{1}{2N(\mathbf{h})} \sum_{\alpha=1}^{N(h)} [z(\mathbf{u}_\alpha + \mathbf{h}) - z(\mathbf{u}_\alpha)]^2. \tag{13}$$

When the distance $|\mathbf{h}|$ is sufficiently large, there may be no spatial correlation between data values at two locations separated by $|\mathbf{h}|$. A contour plot of the experimental semivariogram, called the *variogram surface,* may be used to visualise the spatial continuity in all directions.

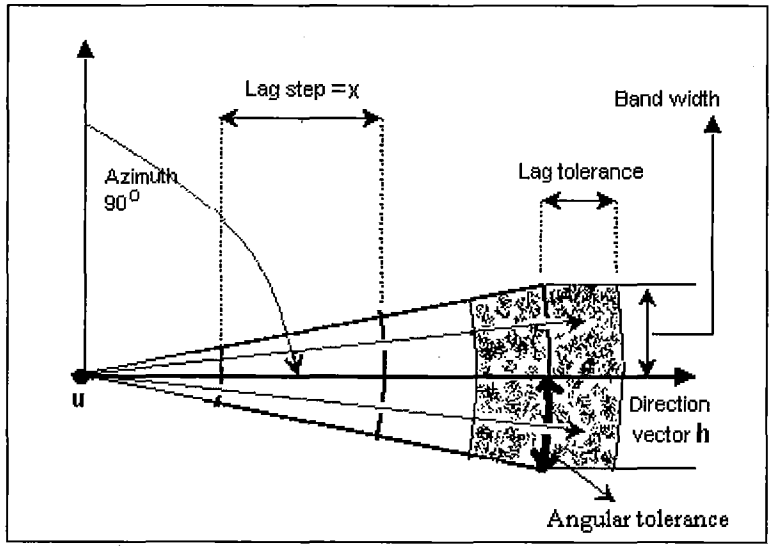


Figure 1: Pairing rule for irregular data in the direction of azimuth angle 90^0 (modified from Deutsch and Journel, 1998).

In practice, for irregularly-spaced data a certain lag tolerance and a certain angular tolerance are applied to ensure that there will be sufficient data pairs. In order to exclude points that lie in areas too far off the search direction, a bandwidth is introduced. The bandwidth is the maximum acceptable deviation in the direction perpendicular to the direction of vector $\mathbf{h}$. For example, in Figure 1 all data lying within the fan are regarded as being in the direction of vector $\mathbf{h}$ (azimuth angle 90^0).

In calculating the spatial correlation in that direction with the distance equal to three lag steps ($3x$), all sample points lying in the shaded region are the associated heads of point u.

Having obtained the experimental semivariogram (covariance, correlation) of the sample data, a model is fitted to these measurements. Since, when it exists, the covariance function of a random function can be obtained from the semivariogram by (9), in this thesis we discuss only the modelling of the semivariogram.

2.3 Semivariogram Models

In order to interpolate the unknown values, we need a semivariogram function $\gamma(\mathrm{h})$ that shows the dissimilarity for any possible separation vector between data. Since the experimental semivariogram only gives values at discrete lags, a smooth curve is fitted to the experimental semivariogram. This function is called the semivariogram model.

2.3.1 Permissibility Condition

Some estimation methods require computations that make use of the variance of some linear combination of random variables. Hence, the semivariogram model must ensure the non-negativity of the variance. Let Z be a stationary random function in a region A specified by the covariance $C(\mathrm{h})$. Suppose that Y is a linear combination of the random variables $Z(\mathrm{u}_\alpha), \alpha = 1, 2, ..., n$, then Y is a random variable and its variance

$$Var(Y) = Var\left\{\sum_{\alpha=1}^{n} \lambda_\alpha Z(\mathrm{u}_\alpha)\right\} \tag{14}$$

must be non-negative. The expression (14) can be rewritten as

$$Var(Y) = \sum_{\alpha=1}^{n}\sum_{\beta=1}^{n} \lambda_\alpha \lambda_\beta C(\mathrm{u}_\alpha - \mathrm{u}_\beta)$$

or equivalently in matrix form as

$$Var(Y) = \begin{pmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_n \end{pmatrix} C_{\alpha\beta} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_n \end{pmatrix}, \alpha, \beta = 1, 2, \dots, n. \tag{15}$$

In order for the variance to be non-negative, the covariance matrix $\{C_{\alpha\beta}\}_{\alpha,\beta=1,2,\dots,n}$ must be positive definite.

Using the relation $\gamma(\mathrm{h}) = C(0) - C(\mathrm{h})$, the variance in (14) can also be written as

$$Var(Y) = C(0) \sum_{\alpha=1}^{n} \lambda_\alpha \sum_{\beta=1}^{n} \lambda_\beta - \sum_{\alpha=1}^{n} \sum_{\beta=1}^{n} \lambda_\alpha \lambda_\beta \gamma(\mathrm{u}_\alpha - \mathrm{u}_\beta) \geq 0. \tag{16}$$

In the case that $C(0)$ is not finite, the value of $C(0)$ is filtered by letting

$$\sum_{\alpha=1}^{n} \lambda_\alpha = 0. \tag{17}$$

In this case the expression (16) becomes

$$Var(Y) = -\sum_{\alpha=1}^{n} \sum_{\beta=1}^{n} \lambda_\alpha \lambda_\beta \gamma(\mathrm{u}_\alpha - \mathrm{u}_\beta) \geq 0. \tag{18}$$

With the condition in (17), the variance of random variable Y is non-negative if the semivariogram model is negative definite. The semivariogram model γ is then said to be *conditionally negative definite* since it is only negative definite with the condition in (17).

2.3.2 Permissible Models

A simple way to avoid checking the permissibility condition is to limit the semivariogram model to functions that are known to be negative definite. In practice a

semivariogram model is usually chosen to be a linear combination of one or more of the five permissible models described below. These models are expressed here in their isotropic form and are classified into *bounded* and *unbounded* models. Both bounded and unbounded models are monotonic. However, a bounded semivariogram model approaches a plateau called the *sill* and the distance at which the sill is approached is called the *range* (Isaaks and Srivastava, 1989), whereas an unbounded model has no sill and hence no range.

Nugget effect model A nugget effect model is a bounded model used to model phenomena that show an obvious discontinuity at the origin. The semivariogram has value 0 at $\mathbf{h} = 0$, then takes a non-zero, positive value at any distance greater than zero. This standardised model (the sill value is scaled to 1) is given by

$$g(|\mathbf{h}|) = \begin{cases} 0 & \text{if } |\mathbf{h}| = 0 \\ 1 & \text{otherwise} \end{cases} . \tag{19}$$

For a nugget model, the sill is reached as soon as the distance is greater than 0.

Spherical model The spherical model has linear behaviour near the origin; it is a commonly used bounded model. Its standardised form is

$$g(|\mathbf{h}|) = \begin{cases} 1.5\frac{|\mathbf{h}|}{a} - 0.5\left(\frac{|\mathbf{h}|}{a}\right)^3 & \text{if } |\mathbf{h}| \leq a \\ 1 & \text{otherwise} \end{cases} \tag{20}$$

where a denotes the range which is the distance at which the semivariogram model reaches the sill.

Exponential model Another commonly used bounded semivariogram model is the exponential model which also has linear behaviour near the origin. An exponential

semivariogram model approaches the sill asymptotically and its formula is given by

$$g(|\mathbf{h}|) = 1 - \exp\left(\frac{-3|\mathbf{h}|}{a}\right) \tag{21}$$

where a is the *practical range*, the distance at which the semivariogram reaches 95% of the sill value.

Gaussian model The Gaussian model is a bounded model that has parabolic behaviour near the origin. It is given by

$$g(|\mathbf{h}|) = 1 - \exp\left(\frac{-3|\mathbf{h}|^2}{a^2}\right). \tag{22}$$

Similar to the exponential model, a Gaussian semivariogram model approaches the sill asymptotically; its practical range is the distance where the semivariogram reaches 95% of the sill value. The spherical, exponential and Gaussian models are illustrated in Figure 2.

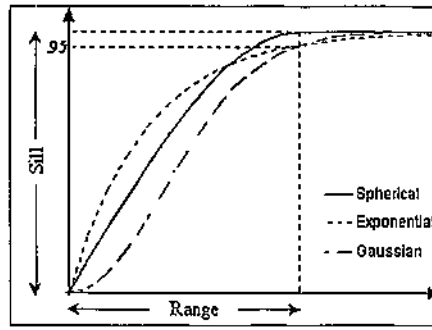


Figure 2: Bounded semivariogram models (modified from Goovaerts, 1997).

Power model The power model is an unbounded model given by

$$g(|\mathbf{h}|) = |\mathbf{h}|^{\omega} \text{ with } 0 < \omega < 2. \tag{23}$$

It has no sill and hence no range. The behaviour near the origin depends on the exponent ω. It has linear behaviour when $\omega = 1$ and parabolic behaviour as ω approaches 2. This model is illustrated in Figure 3 for three specific values of ω.

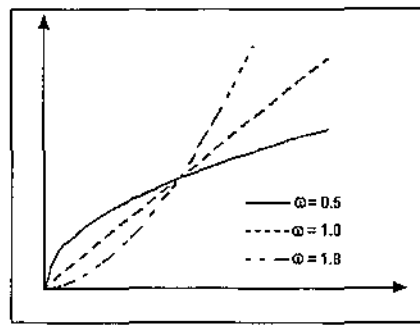


Figure 3: Power model with three values of ω (modified from Goovaerts, 1997).

Anisotropic semivariogram models As mentioned in Section 2.2, an experimental semivariogram is said to be isotropic if it depends only on the distance $|\mathbf{h}|$. In the case when the semivariogram varies with direction, it is said to be anisotropic. Depending upon on the nature of the variation, anisotropic semivariograms are classified into two types: those with geometric anisotropy and those with zonal anisotropy.

Geometric anisotropy When the sill value of the semivariogram is constant but the range varies with direction we have geometric anisotropy. In two-dimensional space, geometric anisotropy can be identified by means of a rose diagram, which is the plot of the range values against the azimuth θ, by convention, measured in degrees clockwise from North. In two-dimensional space, if the rose diagram is an ellipse then the direction of the major axis is the direction of maximum continuity and has the

longest range. The direction of minimum continuity is the direction of the minor axis. Figure 4 gives an example of geometric anisotropy. In Figure 4A, the sill value of the spherical semivariogram models $g(h_\theta)$ and $g(h_\phi)$ in the direction of azimuths θ and ϕ, respectively, are identical. However, the range values a_θ and a_ϕ are different. The associated rose diagram in Figure 4B is an ellipse with the direction of the major axis in the direction of azimuth θ; the major radius is equal to the maximum range a_θ and the minor radius is equal to the minimum range a_ϕ. As all permissible semivariogram

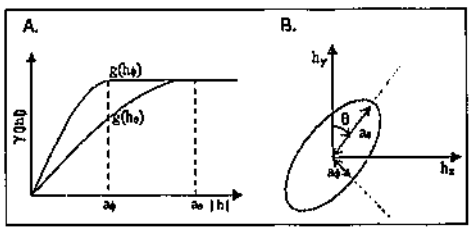


Figure 4: Geometric anisotropy: A/. Semivariogram models; B/. The rose diagram, modified from Goovaerts (1997).

models introduced previously are in isotropic form, a coordinate transformation has to be carried out to express the model that exhibits geometric anisotropy in isotropic form with respect to the new coordinates. The process is accomplished by, firstly, rotating the coordinate axes clock-wise so that the positive direction of the y-axis is identical with the major axis of the ellipse. Secondly, the $y-$coordinates in the new system are rescaled by a value equal to the anisotropy factor $\lambda = a_\phi/a_\theta$. The transformation can be written as

$$\mathbf{h}' = D_\lambda Rot_\theta \mathbf{h}.$$

where

$$D_\lambda = \begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix} \text{ and } Rot_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Zonal anisotropy When the sill value of the semivariogram varies with direction we have zonal anisotropy. In this case, the sill value in the direction of minimum continuity is higher than that in the direction of maximum continuity. Let us denote by ϕ the azimuth angle specifying the direction that has the highest sill value; zonal anisotropy can be modelled as a sum of an isotropic model g_1 and a zonal component g_2 as follows

$$g(\mathrm{h}) = g_1(|\mathrm{h}|) + g_2(h_\phi). \tag{24}$$

Here h_ϕ is the distance measured in the direction that has the highest sill value. The zonal component is computed as follows. Firstly, the coordinate axes are rotated clock-wise so that the y-axis represents the direction of maximum continuity. Secondly, the y—coordinates in the new system are rescaled by setting the range in the direction of maximum continuity to a very large value so that the zonal component depends only on the distance in the direction of minimum continuity. Consequently, the anisotropy factor λ approaches 0 and the zonal component depends only on h_ϕ. Figure 5 gives an illustration for the semivariogram models in the case of zonal anisotropy where $g_1(|\mathrm{h}|)$ denotes the semivariogram model in the direction of maximum continuity and $g_1(|\mathrm{h}|)+g_2(h_\phi)$ denotes the semivariogram model in the direction of minimum spatial continuity.

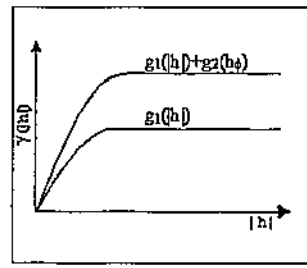


Figure 5: Zonal anisotropy semivariogram models (modified from Goovaerts, 1997).

2.4 Kriging

Kriging is an estimation method used to interpolate the unknown values in a region using a sample of n data $\{z(\mathbf{u}_\alpha), \alpha = 1, 2, ..., n\}$. It was developed by Matheron and Krige (Journel and Huijbregts, 1978). There are many types of kriging, both linear and non-linear. Only linear kriging is discussed since it is the version used in the simulation algorithms we consider. The basic principle of linear kriging is that the value at each unsampled location can be estimated from a subset of the values at the sampled locations using linear regression algorithms. Suppose that $\{z(\mathbf{u}_\alpha), \alpha = 1, 2, ..., n(\mathbf{u})\}$ are the known values of the attribute in a suitably chosen neighbourhood of a location $\mathbf{u}$ where the value $z(\mathbf{u})$ needs to be estimated. We denote by $Z_K^*(\mathbf{u})$ the kriging estimate of the value $z(\mathbf{u})$. In terms of random variables the estimator is given by

$$Z_K^*(\mathbf{u}) - m(\mathbf{u}) = \sum_{\alpha=1}^{n(\mathbf{u})} \lambda_\alpha(\mathbf{u}_\alpha)[Z(\mathbf{u}_\alpha) - m(\mathbf{u}_\alpha)] \tag{25}$$

where $m(\mathbf{u})$ denotes the expected value of the random variable $Z(\mathbf{u})$ at the location $\mathbf{u}$. The values $\lambda_\alpha(\mathbf{u}_\alpha)$ are the weights to be determined so that the expected value of the estimation error is equal to 0 and the variance of the error is minimised. In other

words, the weights $\lambda_\alpha(\mathbf{u}_\alpha)$ are calculated by minimising

$$\sigma_E^2(\mathbf{u}) = Var[Z_K^*(\mathbf{u}) - Z(\mathbf{u})] \tag{26}$$

with the constraint that

$$E\{Z_K^*(\mathbf{u}) - Z(\mathbf{u})\} = 0. \tag{27}$$

In *simple kriging*, the random function Z is assumed to be second-order stationary with a known constant mean m. Therefore, from (25), we have

$$Z_{SK}^*(\mathbf{u}) = \sum_{\alpha=1}^{n(\mathbf{u})} \lambda_\alpha(\mathbf{u}) Z(\mathbf{u}_\alpha) + [1 - \sum_{\alpha=1}^{n(\mathbf{u})} \lambda_\alpha(\mathbf{u})]m \tag{28}$$

where the weights $\lambda_\alpha(\mathbf{u}_\alpha)$ are determined by solving the system of $n(\mathbf{u})$ linear equations:

$$\sum_{\beta=1}^{n(\mathbf{u})} \lambda_\beta(\mathbf{u}) C(\mathbf{u}_\alpha - \mathbf{u}_\beta) = C(\mathbf{u}_\alpha - \mathbf{u}) \qquad \alpha = 1, ..., n(\mathbf{u}). \tag{29}$$

For *ordinary kriging*, the random function Z is only required to be intrinsic stationary, and (28) becomes

$$Z_{OK}^*(\mathbf{u}) = \sum_{\alpha=1}^{n(\mathbf{u})} \lambda_\alpha(\mathbf{u}) Z(\mathbf{u}_\alpha) + [1 - \sum_{\alpha=1}^{n(\mathbf{u})} \lambda_\alpha(\mathbf{u})]m^*(\mathbf{u}), \tag{30}$$

where $m^*(\mathbf{u})$ is assumed to be constant in a neighbourhood and usually unknown. Hence, for the unbiasedness constraint in (27) to be satisfied, the weights $\lambda_\alpha(\mathbf{u}_\alpha)$ are determined by solving the system of $(n(\mathbf{u}) + 1)$ linear equations stated below:

$$\begin{aligned} \sum_{\beta=1}^{n(\mathbf{u})} \lambda_\beta(\mathbf{u}) C(\mathbf{u}_\alpha - \mathbf{u}_\beta) + \mu(\mathbf{u}) &= C(\mathbf{u}_\alpha - \mathbf{u}), \qquad \alpha = 1, ..., n(\mathbf{u}) \\ \sum_{\beta=1}^{n(\mathbf{u})} \lambda_\beta(\mathbf{u}) &= 1 \end{aligned}$$

where $\mu(\mathbf{u})$ is the Lagrange parameter, and the equation $\sum_{\beta=1}^{n(\mathbf{u})} \lambda_\beta(\mathbf{u}) = 1$ is the unbiasedness condition.

Both simple and ordinary kriging estimators are exact interpolators. The estimate at a sample location is equal to the sample value

$$Z_K^*(\mathbf{u}_\alpha) = z(\mathbf{u}_\alpha) \tag{31}$$

and the kriging estimate at an unsampled location, for a given covariance model, is unique.

Kriging is a best linear unbiased estimation (BLUE) method. It is best in the sense that the variance of the estimation error is minimised. It is unbiased since the expected value of the estimation error is equal to zero. However, due to the method of estimation, kriging does not reproduce the histogram and the variance of the sample data. In addition, because of the unbiasedness condition, the estimation by ordinary kriging also exhibits conditional bias. The locations with high attribute values are underestimated and the locations with low attribute values are overestimated. Therefore, the map of the estimates is always smoother than the true map.

In this thesis, simple kriging is used in the sequential Gaussian simulation described in the following section. Ordinary kriging is used in the posprocessing step used to obtain conditional simulated realisations in the simulation algorithm PWSIM described in Chapter 7.

2.5 Simulation

The aim of geostatistical simulation techniques is to apply the Monte Carlo simulation method to generate realisations $z_s(\mathbf{u})$ that reflect the properties of a random function $\mathcal{Z}$. This random function is usually specified by the expectation m and the covariance $C(\mathbf{h})$ which is inferred from the sample covariance computed from the associated sample $\{z(\mathbf{u}_\alpha), \alpha = 1, 2, ..., n\}$. Two types of simulation are distinguished: if the

values at the sample locations in the simulated realisations are equal to the actual data, that is

$$z_s(\mathbf{u}_\alpha) = z(\mathbf{u}_\alpha) \tag{32}$$

for $\alpha = 1, 2, ..., n$, the simulation is said to be *conditional*; if condition (32) is not satisfied, the simulation is said to be *nonconditional*.

Conditional simulation is prefered to linear kriging in applications that focus on the fluctuation patterns of extreme values, for example, the pollution of heavy metals. Because of the smoothing effect, linear kriging may underestimate the presence of high pollutant values leading to the inability to recognise potential health threats. Conditional simulation is also used to measure the spatial uncertainty of an attribute in risk analysis or in decision making. For example, in order to evaluate the risk of classifying a region as safe with respect to some type of pollution, one can generate a set of conditional realisations then the risk can be visualised via the plots of these realisations or can be numerically quantified via a mathematical model. In this thesis we focus only on multiGaussian simulation algorithms, that is, the underlying random function is assumed to have a multivariate normal distribution.

Conditional simulations can be obtained by using linear kriging to postprocess realisations simulated via a non-conditional simulation algorithm, for example, the turning band method (Journel and Huijbregts, 1978), the spectral simulation method (Chilès and Delfiner, 1999 and Pardo-Igúzquiza and Chica-Olmo, 1993) or by using a conditional simulation algorithm. In the first case, at each grid node the simulated value is conditioned by the following steps. Firstly, obtain two kriging estimates at this location, one using the actual data and one using the nonconditional simulated values at data locations, respectively. Secondly, compute the difference of the two kriging values. Finally, add this error to the nonconditional simulated value at this location.

That is, suppose that $z_s(\mathbf{u})$ is an unconditional simulated value, the postprocessing method computes the conditional simulated value $z_{sc}(\mathbf{u})$ from $z_s(\mathbf{u})$ by putting

$$z_{sc}(\mathbf{u}) = z_s(\mathbf{u}) + [z_K^*(\mathbf{u}) - z_{sK}^*(\mathbf{u})] \tag{33}$$

where $z_K^*(\mathbf{u})$ is the simple kriging estimate using the actual data and $z_{sK}^*(\mathbf{u})$ is the simple kriging estimate using the simulated values at the data locations (Journel and Huijbregts, 1978). In terms of the random function model, (33) can be rewritten as:

$$Z_{sc}(\mathbf{u}) = Z_s(\mathbf{u}) + [Z_K^*(\mathbf{u}) - Z_{sK}^*(\mathbf{u})]. \tag{34}$$

Since the configuration between data locations as well as the covariance model is the same for the two kriging images, equation (33) only involves solving one kriging system at each location to be simulated.

In the second case, the data are conditioned directly during the simulation. The sequential Gaussian simulation method (Chilès and Delfiner, 1999) is an example. In the sequential Gaussian simulation algorithm (SGSIM), the simulation is carried out sequentially along a random path that visits each location in the study region once and only once. At each location on the random path, the simulated value is drawn from a parametric distribution constructed from the kriging estimate conditional to the original data and previous simulated values. Then the newly simulated value, in turn, becomes one of the conditioning values for all subsequent simulated values. Since the random path defines the order of locations to be simulated in the study region, for each different path a different simulated realisation is obtained. The results of SGSIM depend on the search radius, the choice of the type of kriging estimate, the number of original data and previously simulated data used for simulating a value.

The performance comparison between SGSIM and DWSIM, a multi-level wavelet-based conditional simulation algorithm developed in this thesis, will be carried out in

Chapter 10.

3 WAVELET ANALYSIS

3.1 Wavelet Analysis in One Dimension

In this Section, we first show how wavelets are constructed and then describe the fast algorithm which is used to compute the wavelet and scaling coefficients for a set of discrete values on a regular grid.

3.1.1 Deterministic Wavelet Analysis

Now consider the Hilbert space $L^2(\mathbb{R})$ (see Rudin, 1991) of all square-integrable functions defined on the real line $\mathbb{R}$, that is, a complex-valued function f belongs to $L^2(\mathbb{R})$ if it satisfies

$$\int_{-\infty}^{\infty} |f(t)|^2 dt < \infty. \tag{35}$$

Finitely supported wavelets, denoted by $\{\psi_{j,n}\}_{j,n\in\mathbb{Z}}$, are families of well-localised functions in $L^2(\mathbb{R})$ each member of which takes non-zero values on a finite interval only, and whose integral over its support is equal to zero (Kahane and Lemarie-Rieusset, 1995). These wavelets are obtained by a dilation and a translation of a *mother wavelet* ψ by

$$\psi_{j,n}(t) = 2^{-j/2}\psi(2^{-j}t - n); j, n \in \mathbb{Z}. \tag{36}$$

If a wavelet family $\{\psi_{j,n}\}_{j,n\in\mathbb{Z}}$ is an orthogonal basis for $L^2(\mathbb{R})$, any function f in $L^2(\mathbb{R})$ can be approximated by a linear combination of the functions $\psi_{j,n}$. In the next Section we introduce the concept of multiresolution approximations, the tool that is used in the construction of orthogonal wavelet bases.

Wavelets and Multiresolution Approximations The approximation of a function f in $L^2(\mathbb{R})$ at scale 2^j (or at resolution 2^{-j}) is an orthogonal projection of f

onto the subspace $V_{j;j\in\mathbb{Z}}$ of $L^2(\mathbb{R})$. This approximation is specified if the associated subspace $V_{j;j\in\mathbb{Z}}$ and the basis for the subspace are defined. A *multiresolution approximation* is used to compute the approximation of a function at different scales. It is constructed as follows:

For $\phi \in L^2(\mathbb{R})$, let $\phi_{j,n}$ be the family of functions defined by

$$\phi_{j,n}(t) = 2^{-j/2}\phi(2^{-j}t - n) \tag{37}$$

and let $\{V_j\}_{j\in\mathbb{Z}}$ be the sequence of closed subspaces of all linear combinations of $\phi_{j,n}$ (including their limit under the $L^2(\mathbb{R})$ norm). The sequence $\{V_j\}_{j\in\mathbb{Z}}$ is a *multiresolution approximation* if the following conditions are met (Mallat, 1998):

1. The subspaces are nested:

$$\ldots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots \tag{38}$$

2. For all $(j,n) \in \mathbb{Z}^2$,

$$f(t) \in V_j \Leftrightarrow f(t - 2^j n) \in V_j. \tag{39}$$

3. The closure of the union of all V_j is $L^2(\mathbb{R})$

$$\lim_{j\to-\infty} V_j = \overline{\bigcup_{j=-\infty}^{\infty} V_j} = L^2(\mathbb{R}). \tag{40}$$

4. The intersection of all V_j is the zero subspace

$$\lim_{j\to\infty} V_j = \bigcap_{j=-\infty}^{\infty} V_j = \{0\}. \tag{41}$$

5. The scaling property is satisfied: $f(t) \in V_j \Longleftrightarrow f(t/2) \in V_{j+1}$.

6. There exists a function $\phi \in V_0$ such that $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ is an orthonormal basis of V_0.

In this case, the function ϕ is called the *scaling function* or the *father wavelet*. From conditions 5 and 6, it follows that, for each j, the set $\{\phi_{j,n} : n \in \mathbb{Z}\}$ is an orthonormal basis of the subspace V_j. The space V_j is also called the space of all approximations of the function f at the scale 2^j. Consequently, for a fixed j, a function $f \in L^2(\mathbb{R})$ can be approximated by a function f_j in V_j by

$$f_j(t) = \sum_{n=-\infty}^{\infty} c_{j,n}\phi(2^{-j}t - n).$$

Since $V_j \subset V_{j-1}$, the scaling function ϕ satisfies

$$\phi(t) = \sqrt{2} \sum_{n=-\infty}^{\infty} h[n]\phi(2t - n), \tag{42}$$

with

$$h[n] = \sqrt{2} \int_{-\infty}^{\infty} \phi(t)\overline{\phi(2t - n)}dt, \tag{43}$$

where the over-bar stands for complex conjugation and by construction (Daubechies, 1988)

$$\sum_{n\in\mathbb{Z}} h[n] = \sqrt{2}. \tag{44}$$

The next step is to show how orthogonal wavelets are defined using the multiresolution approximation. In $L^2(\mathbb{R})$ we define a function ψ by

$$\psi(t) = \sqrt{2} \sum_{n\in\mathbb{Z}} g[n]\phi(2t - n), \tag{45}$$

where

$$g[n] = (-1)^n h[1 - n]. \tag{46}$$

Then it can be proved (Mallat, 1998) that for a fixed j, the family of functions $\{\psi_{j,n}, n \in \mathbb{Z}\}$ defined by

$$\psi_{j,n}(t) = 2^{-j/2}\psi(2^{-j}t - n)$$

is a basis of the orthogonal complement W_j of V_j and these subspaces satisfy

$$V_{j-1} = V_j \oplus W_j, \tag{47}$$

where $\oplus$ denotes the direct sum. The subspace W_j is also called the *detail space*. For all integers j and k where $j \neq k$, the subspaces W_j and W_k are mutually orthogonal and their direct sum is dense in $L^2(\mathbb{R})$, that is

$$L^2(\mathbb{R}) = ... \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus ... \tag{48}$$

The family of functions $\{\psi_{j,n}, j, n \in \mathbb{Z}\}$ are called *wavelets* obtained by dilating and shifting the *mother wavelet* ψ and these functions form a basis for $L^2(\mathbb{R})$.

Definition 1 *The function $h[n]$ in equation (43) is called the lowpass filter and the function $g[n]$ in equation (46) is called the highpass filter.*

Examples of Orthogonal Finitely Supported Wavelets The following sections give examples for orthogonal finitely supported wavelets constructed via multiresolution approximations together with their filters. These wavelet families will be used as the bases for the simulation algorithms in this thesis.

Haar Wavelets Haar wavelets are the simplest orthogonal and finitely supported wavelets. They are obtained by the dilation and translation of the Haar mother wavelet ψ_H, where

$$\psi_H(t) = \begin{cases} 1 & 0 \leq t < 0.5 \\ -1 & 0.5 \leq t < 1 \\ 0 & otherwise \end{cases}. \tag{49}$$

The Haar mother wavelet is constructed from the Haar father wavelet ϕ_H given by

$$\phi_H(t) = \begin{cases} 1 & 0 \leq t < 1 \\ 0 & otherwise \end{cases} \tag{50}$$

with lowpass filter

$$h[n] = \begin{cases} \sqrt{2}/2 & n = 0, 1 \\ 0 & otherwise \end{cases}$$

and highpass filter

$$g[n] = \begin{cases} \sqrt{2}/2 & n = 0 \\ -\sqrt{2}/2 & n = 1 \\ 0 & otherwise \end{cases}.$$

The Haar mother wavelet is the only orthogonal and finitely supported wavelet that has a closed form.

Daubechies wavelets Daubechies wavelets are orthogonal and finitely supported functions that have no explicit formulae but are characterised by their lowpass and highpass filters. Since they have finite support, their filters have finite length. The number of non-zero coefficients of the filters depends on the number of vanishing moments of the wavelets.

Definition 2 *Let $M \geq 2$ be a positive integer. A wavelet family is said to have M vanishing moments if*

$$\int_{-\infty}^{\infty} t^k \psi(t) dt = 0; k = 0, ..., M-1.$$

For wavelets with M vanishing moments, the associated filters have $2M$ non-zero coefficients. Haar wavelets can be considered as wavelets with one vanishing moment.

The more non-zero coefficients the filters have, the smoother the Daubechies mother wavelet and scaling functions become. The values of the filters are determined using the properties and restrictions of multiresolution approximations and orthonormality (see Daubechies, 1988). Depending on the number of non-zero coefficients of the filters, we have different families of Daubechies wavelets, for example, Db2 denotes the family of orthogonal and compactly supported wavelets with two vanishing moments; Db3 denotes the Daubechies wavelets with three vanishing moments; Dbn denotes the Daubechies wavelets with n vanishing moments. The filter coefficients of Db2 (correct to 10 decimal places) are listed in Table 1.

n	h[n]	g[n]
0	.4829629131	-.1294095226
1	.8365163037	-.2241438680
2	.2241438680	.8365163037
3	-.1294095226	-.4829629131

Table 1: Db2 filter coefficients

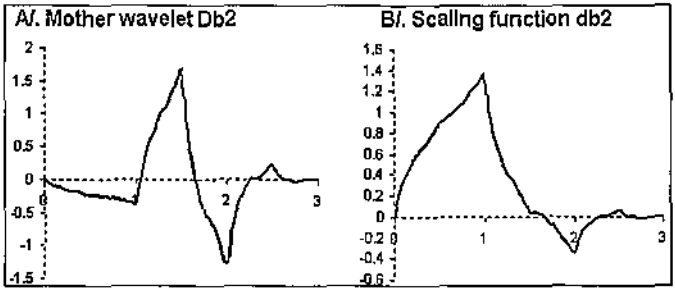


Figure 6: Mother wavelet Db2 and scaling function db2.

Since Daubechies wavelets have no explicit formulae, for each value of t, the associated value of $\psi(t)$ is determined using the "cascade algorithm" (see Daubechies, 1992). Figure 6 shows the plots of the Daubechies mother wavelet Db2 and of the associated scaling function db2 generated via MATLAB.

The Discrete Wavelet Transform and Inverse Discrete Wavelet Transform

The discrete wavelet transform is a fast algorithm that computes the scaling and wavelet coefficients at a coarser scale 2^{j+1} from the scaling coefficients at the finer scale 2^j. The inverse discrete wavelet transform is the operation that reconstructs the scaling coefficients at the finer scale 2^j from the scaling and wavelet coefficients at the coarser scale 2^{j+1}.

For a set of uniformly-spaced values $\{c^0[n]\}_{n\in\mathbb{N}}$ we define a function $f \in V_0$ by

$$f(t) = \sum_{n=-\infty}^{\infty} c^0[n]\phi(t-n). \tag{51}$$

Using the orthonormality property of scaling functions $\{\phi_{0,n}\}_{n\in\mathbb{N}}$, it follows that

$$c^0[n] = \langle f, \phi_{0,n} \rangle \tag{52}$$

where $\langle .,. \rangle$ denotes the inner product in $L^2(\mathbb{R})$ (see Rudin, 1991). The scaling coefficients $c^j[n]$ and wavelet coefficients $d^j[n]$ of the set $\{c^0[n]\}_{n\in\mathbb{N}}$ are defined by

$$c^j[n] = \langle f, \phi_{j,n} \rangle \tag{53}$$

and

$$d^j[n] = \langle f, \psi_{j,n} \rangle . \tag{54}$$

In this light, the set of discrete values $\{c^0[n]\}_{n\in\mathbb{N}}$ is identical with the set of scaling coefficients at scale 2^0 and $d^0[n] = 0$ for all n since f is a function in V_0.

The following theorem (Mallat, 1998) allows the scaling coefficients at scale 2^j to be decomposed into and reconstructed from the scaling and wavelet coefficients at scale 2^{j+1}.

Theorem 3 *Discrete wavelet transform: Each scaling coefficient at the coarser scale 2^{j+1} is computed from the scaling coefficients at the finer scale 2^j by*

$$c^{j+1}[n] = \sum_{p=-\infty}^{\infty} h[p-2n]c^j[p]. \tag{55}$$

Each wavelet coefficient at the coarser scale 2^{j+1} is computed from the scaling coefficients at the finer scale 2^j by

$$d^{j+1}[n] = \sum_{p=-\infty}^{\infty} g[p-2n]c^j[p]. \tag{56}$$

Inverse discrete wavelet transform: Each scaling coefficient at the finer scale 2^j is computed from the scaling and wavelet coefficients at the coarser scale 2^{j+1} by

$$c^j[p] = \sum_{n=-\infty}^{\infty} h[p-2n]c^{j+1}[n] + \sum_{n=-\infty}^{\infty} g[p-2n]d^{j+1}[n]. \tag{57}$$

Here the functions g and h denote respectively the highpass and lowpass filters corresponding to the chosen wavelet basis. Since the number of non-zero coefficients of the filters is $2M$ where M is the number of vanishing moments, putting $u = p-2n$, we may write (55) and (56) as finite sums:

$$c^{j+1}[n] = \sum_{u=0}^{2M-1} h[u]c^j[2n+u] \tag{58}$$

and

$$d^{j+1}[n] = \sum_{u=0}^{2M-1} g[u]c^j[2n+u]. \tag{59}$$

3.1.2 Stochastic Wavelet Analysis

In cases where only some values on the regular grid are known, the wavelet and scaling coefficients at the coarser scale cannot be obtained simply by applying the discrete wavelet transform to the scaling coefficients at the finer scale. Likewise, the scaling coefficients at the finer scale cannot be computed from the scaling and wavelet coefficients at the coarser scale simply using the inverse discrete wavelet transform. Stochastic wavelet analysis is used to handle such situations.

In stochastic wavelet analysis, the attribute value at each grid node is regarded as a particular realisation of a random variable at this location. Then the scaling and wavelet coefficients, which are linear combinations of random variables, are also random variables. Under the assumptions that the underlying random function Z is second-order stationary and that it follows a multivariate standard normal distribution with covariance function $C(|\mathbf{h}|)$, the scaling coefficients and wavelet coefficients also follow a normal distribution with mean zero. In addition, if the attribute values are correlated, the scaling and wavelet coefficients are also correlated. The correlation of scaling and wavelet coefficients can be used to simulate the unknown scaling and wavelet coefficients from the known ones.

Covariance between Two Scaling Coefficients at the Same Scale Since the scaling coefficients $c^0[.]$ at the original scale 2^0 are identical with the correlated original values, the scaling coefficients at scale $2^j, j > 0$ are also correlated. Let us denote by $\beta^j_{(m,n)}$ the covariance between two scaling coefficients $c^j[m]$ and $c^j[n], j > 0$. Then

using equation (58) and $E\{c^j[.]\} = 0$, we can write

$$\beta^j_{(m,n)} = E\{c^j[m]c^j[n]\} - E\{c^j[m]\}E\{c^j[n]\} \tag{60}$$

$$= E\{\sum_{u=0}^{2M-1} h[u]c^{j-1}[2m+u] \sum_{v=0}^{2M-1} h[v]c^{j-1}[2n+v]\} \tag{61}$$

$$= \sum_{u,v=0}^{2M-1} h[u]h[v]E\{c^{j-1}[2m+u]c^{j-1}[2n+v]\} \tag{62}$$

and so

$$\beta^j_{(m,n)} = \sum_{u,v=0}^{2M-1} h[u]h[v]\beta^{j-1}_{(2m+u,2n+v)}. \tag{63}$$

It can be seen from equation (63) that the covariance between two scaling coefficients at the same scale is a linear combination of the covariance of the scaling coefficients at the previous finer scale.

Covariance between Two Scaling Coefficients at Two Consecutive Scales

In the previous Section we have shown that the covariance between two scaling coefficients at a coarse scale can be computed from the covariance of scaling coefficients at the finer scale. In this Section we will give the formula for computing the covariance of two scaling coefficients at two consecutive scales. Denote by

$$\delta^j_{(m,n)} = E\{c^j[m]c^{j+1}[m]\} - E\{c^j[m]\}E\{c^{j+1}[n]\} \tag{64}$$

the covariance of two scaling coefficients $c^j[m]$ and $c^{j+1}[n]$ where $j \geq 0$; with (58), equation (64) becomes

$$\delta^j_{(n,m)} = E\{c^j[m] \sum_{u=0}^{2M-1} h[u]c^j[2n+u]\}$$

$$= \sum_{u=0}^{2M-1} h[u]E\{c^j[m]c^j[2n+u]\}.$$

According to the definition in (60), the covariance between two scaling coefficients at two consecutive scales is a linear combination of the covariance between scaling

coefficients at the finer scale 2^j and is given by

$$\delta^j_{(m,n)} = \sum_{u=0}^{2M-1} h[u]\beta^j_{(m,2n+u)}. \tag{65}$$

Covariance between Two Wavelet Coefficients at the Same Scale The covariance of two wavelet coefficients $d^j[m]$ and $d^j[n], j > 0$ is denoted by

$$\tau^j_{(m,n)} = E\{d^j[m]d^j[n]\} - E\{d^j[m]\}E\{d^j[n]\}. \tag{66}$$

Using equation (59), the covariance $\tau^j_{(m,n)}$ is computed in terms of the scaling coefficients at the finer scale 2^{j-1} by

$$\begin{aligned} E\{d^j[m]d^j[n]\} &= E\{\sum_{u=0}^{2M-1} g[u]c^{j-1}[2m+u] \sum_{v=0}^{2M-1} g[v]c^{j-1}[2n+v]\} \\ &= \sum_{u,v=0}^{2M-1} g[u]g[v]E\{c^{j-1}[2m+u]c^{j-1}[2n+v]\}. \end{aligned}$$

So the covariance of two wavelet coefficients at scale 2^j is a linear combination of the covariance of scaling coefficients at the finer scale 2^{j-1} and is given by

$$\tau^j_{(m,n)} = \sum_{u,v=0}^{2M-1} g[u]g[v]\beta^{j-1}_{(2m+u,2n+v)}. \tag{67}$$

Covariance between a Wavelet Coefficient and a Scaling Coefficient at the Same Scale The covariance of a wavelet coefficient $d^j[m]$ and a scaling coefficient $c^j[n], j > 0$ is denoted by

$$\eta^j_{(m,n)} = E\{d^j[m]c^j[n]\} - E\{d^j[m]\}E\{c^j[n]\}. \tag{68}$$

Using equations (58) and (59), the covariance $\eta^j_{(m,n)}$ is computed in terms of the scaling coefficients at the finer scale 2^{j-1} by

$$\begin{aligned} E\{d^j[m]c^j[n]\} &= E\{\sum_{u=0}^{2M-1} g[u]c^{j-1}[2m+u] \sum_{v=0}^{2M-1} h[v]c^{j-1}[2n+v]\} \\ &= \sum_{u,v=0}^{2M-1} g[u]h[v]E\{c^{j-1}[2m+u]c^{j-1}[2n+v]\}. \end{aligned}$$

In terms of the covariance of scaling coefficients at the finer scale, the covariance of a wavelet coefficient and a scaling coefficient at the same scale is computed by

$$\eta^j_{(m,n)} = \sum_{u,v=0}^{2M-1} g[u]h[v]\beta^{j-1}_{(2m+u,2n+v)}. \tag{69}$$

For stationary processes, the values β^j, δ^j, $\tau^j$and η^j are independent of the indices of the scaling/wavelet coefficients. The following proposition verifies this property.

Proposition 4 *If the random function $Z : \mathcal{A} \to \{Z(u) : u \in \mathcal{A}\}$ is a second-order stationary Gaussian random function with zero mean then the values $\beta^j, \delta^j, \eta^j$ and τ^j do not depend on the indices of the scaling / wavelet coefficients.*

Proof. We prove Proposition (4) for the case of the covariance of two scaling coefficients only. The same argument is used for other cases. This proof is based on the Spectral Representation Theorem (see Appendix 12.1).

Let $c^{j_1}[k]$ be the scaling coefficient at scale 2^{j_1} and $c^{j_2}[m]$ be the scaling coefficient at scale 2^{j_2}.

From equation (53), the scaling coefficient $c^{j_1}[k]$ is the inner product of Z and the scaling function $\phi_{j_1,k}$, and because $\phi_{j_1,k}$ is a real-valued function the scaling coefficient $c^{j_1}[k]$ is given by

$$c^{j_1}[k] = \int_{-\infty}^{\infty} Z(u)\overline{\phi_{j_1,k}(u)}du = \int_{-\infty}^{\infty} Z(u)\phi_{j_1,k}(u)du. \tag{70}$$

According to the first result of the Spectral Representation Theorem, the value $Z(u)$ can be written as

$$Z(u) = \int_{-\infty}^{\infty} e^{i\omega_1 u} d\dot{Z}(\omega_1) \tag{71}$$

where $\dot{Z}$ denotes a complex random function. By substituting the value $Z(u)$ in (71) into equation (70) we have

$$c^{j_1}[k] = \int_{-\infty}^{\infty} (\int_{-\infty}^{\infty} e^{i\omega_1 u} d\dot{Z}(\omega_1))\phi_{j_1,k}(u)du = \int_{-\infty}^{\infty} (\int_{-\infty}^{\infty} e^{i\omega_1 u}\phi_{j_1,k}(u)du) d\dot{Z}(\omega_1)$$

or

$$c^{j_1}[k] = \int_{-\infty}^{\infty} \overline{\widehat{\phi}_{j_1 k}(\omega_1)} d\dot{Z}(\omega_1) \tag{72}$$

where $\widehat{\phi}_{j_1 k}$ is the Fourier transform (see Mallat, 1995, page 23) of the function $\phi_{j_1 k}$ defined by

$$\widehat{\phi}_{j_1 k}(\omega_1) = \int_{-\infty}^{\infty} e^{-i\omega_1 u} \phi_{j_1 k}(u) du.$$

Similarly, the scaling coefficient $c^{j_2}[m]$ can be expressed as

$$c^{j_2}[m] = \int_{-\infty}^{\infty} \overline{\widehat{\phi}_{j_2 m}(\omega_2)} d\dot{Z}(\omega_2). \tag{73}$$

Since the scaling coefficients are real, equation (73) can be rewritten by

$$c^{j_2}[m] = \overline{c^{j_2}[m]} = \overline{\int_{-\infty}^{\infty} \overline{\widehat{\phi}_{j_2 m}(\omega_2)} d\dot{Z}(\omega_2)} = \int_{-\infty}^{\infty} \widehat{\phi}_{j_2 m}(\omega_2) \overline{d\dot{Z}(\omega_2)}.$$

Next, by computing the covariance of the two scaling coefficients, we have

$$E\{c^{j_1}[k] c^{j_2}[m]\} = E\{\int_{-\infty}^{\infty} \overline{\widehat{\phi}_{j_1 k}(\omega_1)} d\dot{Z}(\omega_1) \int_{-\infty}^{\infty} \widehat{\phi}_{j_2 m}(\omega_2) \overline{d\dot{Z}(\omega_2)}\}$$

$$= E\{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \overline{\widehat{\phi}_{j_1 k}(\omega_1)} \widehat{\phi}_{j_2 m}(\omega_2) d\dot{Z}\omega_1 \overline{d\dot{Z}(\omega_2)}\}. \tag{74}$$

According to the third result of the Spectral Representation Theorem (Appendix 12.1), we have from (74)

$$E\{c^{j_1}[k] c^{j_2}[m]\} = \int_{-\infty}^{\infty} \overline{\widehat{\phi}_{j_1 k}(\omega)} \widehat{\phi}_{j_2 m}(\omega) S(\omega) d\omega \tag{75}$$

where

$$S(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} C(h) e^{-i\omega h} dh.$$

Using the properties of Fourier transform together with the definition of the function $\phi_{j_1 k}$ we have

$$\overline{\widehat{\phi}_{j_1 k}(\omega)} = 2^{j_1/2} e^{i\omega 2^{j_1} k} \overline{\widehat{\phi}(2^{j_1}\omega)} \tag{76}$$

and

$$\widehat{\phi}_{j_2 m}(\omega) = 2^{j_2/2} e^{-i\omega 2^{j_2} m} \widehat{\phi}(2^{j_2}\omega). \tag{77}$$

By substituting the values on the right hand side of equations (76) and (77) into equation (75) and simplifing the result, we have

$$E\{c^{j_1}[k]c^{j_2}[m]\} = \int_{-\infty}^{\infty} 2^{(j_1+j_2)/2} e^{i\omega 2^{j_1}(k-2^{j_2-j_1}m)} \overline{\widehat{\phi}(2^{j_1}\omega)}\widehat{\phi}(2^{j_2}\omega) S(\omega) d\omega.$$

It can be seen that the covariance between the two scaling coefficients $c^{j_1}[k]$ and $c^{j_2}[m]$ does not depend on the integer k and m but only on the different $(k - 2^{j_2-j_1}m)$. ■

We have proved that the covariance of two scaling coefficients $c^{j_1}[k]$ and $c^{j_2}[m]$ depends on $(k - 2^{j_2-j_1}m)$. In particular, the covariance of two scaling coefficients at the same scale $c^j[k]$ and $c^j[m]$ depends only on the difference of the indices k and m.

3.2 Wavelet Analysis in Two Dimensions

In this Section, the concepts used in deterministic and stochastic wavelet analysis in one-dimensional space are extended to two-dimensional space.

3.2.1 Multiresolution Approximations of $L^2(\mathbb{R}^2)$ and Two-Dimensional Wavelet Bases

In two-dimensional space, wavelets are defined as products of one-dimensional wavelet and scaling functions. If ϕ and ψ are the one-dimensional scaling function and mother wavelet, respectively, then the two-dimensional mother wavelets are defined by

$$\psi^1(x,y) = \phi(x)\psi(y); \psi^2(x,y) = \psi(x)\phi(y); \psi^3(x,y) = \psi(x)\psi(y) \tag{78}$$

and the family $\{\psi^k_{j,m,n}\}$ given by

$$\psi^k_{j,m,n}(x,y) = 2^{-j}\psi^k(2^{-j}x - n, 2^{-j}y - m) \tag{79}$$

with $k = 1, 2, 3$ are functions obtained by the dilations and translation of the mother wavelets ψ^k. The two-dimensional wavelets in (78) are obtained via the *separable multiresolution approximations* in $L^2(\mathbb{R}^2)$ described as follows.

For a fixed value of $j, j \in \mathbb{Z}$, the orthogonal projection of a function f in the Hilbert space $L^2(\mathbb{R}^2)$ onto the subspace V_j^2 of $L^2(\mathbb{R}^2)$ is called the approximation of f at scale 2^j. If for each value of j the space V_j^2 is defined by

$$V_j^2 = V_j \otimes V_j \tag{80}$$

where $\otimes$ denotes the tensor product and $\{V_j\}_{j\in\mathbb{Z}}$ is a multiresolution approximation of $L(\mathbb{R}^2)$ then the set of subspaces $\{V_j^2\}_{j\in\mathbb{Z}}$ is a *separable two-dimensional multiresolution approximation* of $L^2(\mathbb{R}^2)$. As a consequence, for any $j \in \mathbb{Z}$, the collection of the dilations and translations of the scaling function $\phi(x, y) = \phi(x)\phi(y)$ given by

$$\{\phi_{j,m,n}(x, y) = \phi_{j,n}(x)\phi_{j,m}(y) = 2^{-j}\phi_{j,n}(2^{-j}x - n)\phi_{j,m}(2^{-j}y - m)\}_{(m,n)\in\mathbb{Z}^2} \tag{81}$$

forms an orthonormal basis of V_j^2. Similar to the multiresolution approximation in one-dimensional space, the subspace V_{j-1}^2 can be decomposed into two mutually orthogonal subspaces of coarser resolution W_j^2 and V_j^2 by

$$V_{j-1}^2 = W_j^2 \otimes V_j^2. \tag{82}$$

It can be proved (Mallat, 1998) that for a fixed value of j, the set

$$\{\psi_{j,m,n}^1(x, y), \psi_{j,m,n}^2(x, y), \psi_{j,m,n}^3(x, y)\}_{(m,n)\in\mathbb{Z}^2} \tag{83}$$

is an orthonormal basis of W_j^2 and for $j \in \mathbb{Z}$ the set of two dimensional wavelets in (83) forms an orthonormal basis for the space $L^2(\mathbb{R}^2)$.

3.2.2 The Two-Dimensional Discrete Wavelet Transform

The two-dimensional discrete wavelet transform is a fast algorithm, based on the multiresolution approximation of $L^2(\mathbb{R}^2)$ and the two-dimensional wavelet bases, used to

compute the scaling and wavelet coefficients of a set of discrete values $\{c^0[m,n]\}_{(m,n)\in\mathbb{Z}^2}$ on a regular grid.

As a result of the multiresolution approximation of $L^2(\mathbb{R}^2)$, the family of scaling functions $\{\phi_{0,m,n}\}_{(m,n)\in\mathbb{Z}^2}$ is a basis for V_0^2. Then the set of discrete values $\{c^0[m,n]\}_{(m,n)\in\mathbb{Z}^2}$ can be considered the set of scaling coefficients at the scale 2^0 of a function $f \in V_0^2$ defined by

$$c^0[m,n] = \langle f, \phi_{0,m,n} \rangle \tag{84}$$

where

$$f(x,y) = \sum_{m,n=-\infty}^{\infty} c^0[m,n]\phi_{0,m,n}(x,y). \tag{85}$$

and as in the one-dimensional case $\langle .,. \rangle$ denotes the inner product. For $j > 0$ and $k = 1,2,3$, the scaling and wavelet coefficients of the set $\{c^0[m,n]\}_{(m,n)\in\mathbb{Z}^2}$ are defined by

$$c^j[m,n] = \langle f, \phi_{j,m,n} \rangle \tag{86}$$

and

$$d_k^j[m,n] = \langle f, \psi_{k,m,n}^j \rangle \tag{87}$$

respectively.

Using the above properties, it can be proved (Mallat, 1998) that the scaling and wavelet coefficients at location (m,n) at scale 2^{j+1} can be computed from the scaling coefficients at the finer scale 2^j by

$$c^{j+1}[m,n] = \sum_{l=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} h[k-2m]h[l-2n]c^j[k,l], \tag{88}$$

$$d_1^{j+1}[m,n] = \sum_{l=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} g[k-2m]h[l-2n]c^j[k,l], \tag{89}$$

$$d_2^{j+1}[m,n] = \sum_{l=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} h[k-2m]g[l-2n]c^j[k,l], \tag{90}$$

$$d_3^{j+1}[m,n] = \sum_{l=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} g[k-2m]g[l-2n]c^j[k,l], \tag{91}$$

and that the scaling coefficients at scale 2^j can be reconstructed from the scaling and wavelet coefficients at the coarser scale 2^{j+1} by

$$\begin{aligned} c^j[m,n] \;=\; & \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} h[m-2k]h[n-2l]c^{j+1}[k,l] \\ & + \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} g[m-2k]h[n-2l]d_1^{j+1}[k,l] \\ & + \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} h[m-2k]g[n-2l]d_2^{j+1}[k,l] \\ & + \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} g[m-2k]g[n-2l]d_3^{j+1}[k,l]. \end{aligned} \tag{92}$$

The decomposition in (88)-(91) is called the two-dimensional discrete wavelet transform and the reconstruction in (92) is called the inverse discrete wavelet transform.

In image processing the two-dimensional discrete wavelet transform is used to compress images to save computer storage space. As the first step, the discrete wavelet transform is applied to the original image $\{c^0[m,n] : 0 \le m < P, 0 \le n < Q\}$ where $P = 2^{m_1}, Q = 2^{n_1}, m_1, n_1 = 0, 1, 2, \ldots$ to decompose it into one set of scaling coefficients $\{c^1[m,n]\}$ and three sets of wavelet coefficients $\{d_k^1[m,n]\}_{k=1,2,3}$ where $0 \le m < P/2, 0 \le n < Q/2$. The set $\{c^1[m,n]\}$ is the compression of $\{c^0[m,n]\}$. The three sets of wavelet coefficients (associated with three types of wavelets in two-dimensional space) contain the detail lost in the compression. Then the set of scaling coefficients can be further compressed by the discrete wavelet transform. Each move from a finer scale 2^j to the next coarser scale 2^{j+1} leads to a reduction in the size of the scaling image by a factor of four. The process is illustrated in Figure 7 where the

discrete wavelet transform is applied to an original image of size 128 by 128 (Figure 7A). After one application of the discrete wavelet transform, one scaling and three wavelet images of size 64 by 64 at the coarser scale are obtained as shown in Figure 7B.

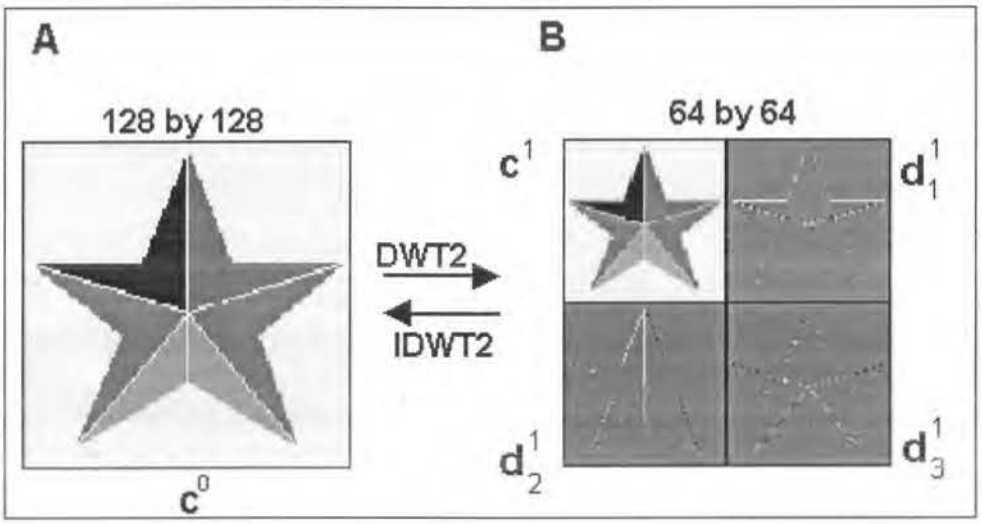


Figure 7: One application of the two-dimensional discrete wavelet transform to an original image of size 128×128.

Similar to the one-dimensional case, putting $u = k - 2m$ and $v = l - 2n$, the equations (88)-(91) can be rewritten as

$$c^{j+1}[m,n] = \sum_{u=0}^{2M-1} \sum_{v=0}^{2M-1} h[u]h[v]c^j[2m+u, 2n+v], \tag{93}$$

$$d_1^{j+1}[m,n] = \sum_{u=0}^{2M-1} \sum_{v=0}^{2M-1} g[u]h[v]c^j[2m+u, 2n+v], \tag{94}$$

$$d_2^{j+1}[m,n] = \sum_{u=0}^{2M-1} \sum_{v=0}^{2M-1} h[u]g[v]c^j[2m+u, 2n+v], \tag{95}$$

$$d_3^{j+1}[m,n] = \sum_{u=0}^{2M-1} \sum_{v=0}^{2M-1} g[u]g[v]c^j[2m+u, 2n+v]. \tag{96}$$

Since in one-dimensional space the sum of the lowpass filter coefficients $\sum_{u=0}^{2M-1} h[u]$ is equal to $\sqrt{2}$, the sum of the lowpass filter coefficients in two dimensional space is equal to 2, that is

$$\sum_{u=0}^{2M-1}\sum_{v=0}^{2M-1} h[u]h[v] = 2. \tag{97}$$

3.2.3 Stochastic Wavelet Analysis in Two-Dimensional Space

In this section, we introduce the formulae for computing the covariance of pairs of scaling coefficients, the covariance between a scaling coefficient and a wavelet coefficient and the covariance of pairs of wavelet coefficients of the same type. These covariances are used in the simulation of realisations of two-dimensional random functions using the algorithms presented in this thesis. In addition, the definition and the computation of the wavelet and scaling spectra are also introduced.

Covariance between a Pair of Scaling Coefficients In two-dimensional space, the covariance between two scaling coefficients at the same scale $c^j[p,q]$ and $c^j[m,n], j > 0$ is denoted by

$$\beta^j_{(p,m),(q,n)} = E\{c^j[p,q]c^j[m,n]\} - E\{c^j[p,q]\}E\{c^j[m,n]\},$$

and the covariance between two scaling coefficients $c^j[p,q]$ and $c^{j+1}[m,n], j \geq 0$ at two consecutive scales is denoted by

$$\delta^j_{(p,m),(q,n)} = E\{c^j[p,q]c^{j+1}[m,n]\} - E\{c^j[p,q]\}E\{c^{j+1}[m,n]\}.$$

Using equation (93) one can prove that the covariance $\beta^j_{(m,p),(n,q)}$ can be computed from the covariance at the previous finer scale $2^{j-1}$by

$$\beta^j_{(p,m),(q,n)} = \sum_{u_1,v_1=0}^{2M-1}\sum_{u_2,v_2=0}^{2M-1} h[u_1]h[v_1]h[u_2]h[v_2]\beta^{j-1}_{(2p+u_1,2m+u_2),(2q+v_1,2n+v_2)}. \tag{98}$$

The covariance $\delta^j_{(p,m),(q,n)}$ between two scaling coefficients at two consecutive scales is computed from the covariance of scaling coefficients at the finer scale by

$$\begin{aligned}\delta^j_{(p,m),(q,n)} &= E\{c^j[p,q]\sum_{u,v=0}^{2M-1} h[u]h[v]c^j[2m+u,2n+v]\} \\ &= \sum_{u=0}^{2M-1} h[u]h[v]E\{c^j[p,q]c^j[2m+u,2n+v]\}.\end{aligned}$$

According to equation (98), we have

$$\delta^j_{(p,m),(q,n)} = \sum_{u,v=0}^{2M-1} h[u]h[v]\beta^j_{(p,2m+u),(q,2n+v)}. \tag{99}$$

Covariance between Two Wavelet Coefficients of the Same Type at the Same Scale In two-dimensional space, there are three types of wavelet covariance. For each k, $k = 1, 2$ or 3 the covariance between two wavelet coefficients $d^j_k[m,n]$ and $d^j_k[p,q], j > 0$ is denoted by

$$\tau^j_{k(m,p),(n,q)} = E\{d^j_k[m,n]d^j_k[p,q]\} - E\{d^j_k[m,n]\}E\{d^j_k[p,q]\}.$$

Using equations (94), (95) and (96), for each value of k, the covariance $\tau^j_{k(m,p),(n,q)}$ is computed in terms of the covariance of scaling coefficients at the finer scale by

$$\tau^j_{1(m,p),(n,q)} = \sum_{u_1,v_1=0}^{2M-1}\sum_{u_2,v_2=0}^{2M-1} h[u_1]g[v_1]h[u_2]g[v_2]\beta^{j-1}_{(2m+u_1,2p+u_2),(2n+v_1,2q+v_2)}, \tag{100}$$

$$\tau^j_{2(m,p),(n,q)} = \sum_{u_1,v_1=0}^{2M-1}\sum_{u_2,v_2=0}^{2M-1} g[u_1]h[v_1]g[u_2]h[v_2]\beta^{j-1}_{(2m+u_1,2p+u_2),(2n+v_1,2q+v_2)}, \tag{101}$$

$$\tau^j_{3(m,p),(n,q)} = \sum_{u_1,v_1=0}^{2M-1}\sum_{u_2,v_2=0}^{2M-1} g[u_1]g[v_1]g[u_2]g[v_2]\beta^{j-1}_{(2m+u_1,2p+u_2),(2n+v_1,2q+v_2)}. \tag{102}$$

Covariance between a Wavelet Coefficient and a Scaling Coefficient at the Same Scale Since there are three types of wavelet coefficients $\{d^j_k[p,q]\}_{k=1,2,3}$ in two-dimensional space, there are also three types of covariance between scaling coefficients and wavelet coefficients associated with each value of k. For $k = 1, 2, 3$, the

covariance between a scaling coefficient $c^j[n,m]$ and a wavelet coefficient $d_k^j[p,q], k = 1,2,3$ the same scale is denoted by

$$\eta^j_{k(m,p),(n,q)} = E\{c^j[n,m]d_k^j[p,q]\}, k = 1,2,3.$$

Using equations (93) - (96), for $k = 1,2,3$ the covariance $\eta^j_{k(m,p),(n,q)}$ is computed in terms of the covariance of scaling coefficients at the finer scale 2^{j-1} by

$$\eta^j_{1(m,p),(n,q)} = \sum_{u_1,v_1=0}^{2M-1}\sum_{u_2,v_2=0}^{2M-1} h[u_1]h[v_1]h[u_2]g[v_2]\beta^{j-1}_{(2m+u_1,2p+u_2),(2n+v_1,2q+v_2)}, \quad (103)$$

$$\eta^j_{2(m,p),(n,q)} = \sum_{u_1,v_1=0}^{2M-1}\sum_{u_2,v_2=0}^{2M-1} h[u_1]h[v_1]g[u_2]h[v_2]\beta^{j-1}_{(2m+u_1,2p+u_2),(2n+v_1,2q+v_2)}, \quad (104)$$

$$\eta^j_{3(m,p),(n,q)} = \sum_{u_1,v_1=0}^{2M-1}\sum_{u_2,v_2=0}^{2M-1} h[u_1]h[v_1]g[u_2]g[v_2]\beta^{j-1}_{(2m+u_1,2p+u_2),(2n+v_1,2q+v_2)}. \quad (105)$$

Similar to the case of one-dimensional space, the covariance $\beta^j, \delta^j, \tau^j_k$ and η^j_k are independent of the indices of the scaling and wavelet coefficients. This property is generalised from Proposition 4 as follows:

Proposition 5 *If $Z : \mathcal{A} \to \{Z(u_1,u_2) : (u_1,u_2) \in \mathcal{A}\}$ is a second-order stationary Gaussian random function with zero mean then the covariances $\beta^j, \delta^j, \tau^j_k$ and η^j_k do not depend on the indices of the scaling and wavelet coefficients.*

Proof. Again we provide the proof for the case of the covariance of two scaling coefficients only. In two-dimensional space, let $c^{j_1}[p,q]$ be the scaling coefficient at scale 2^{j_1} and $c^{j_2}[m,n]$ be the scaling coefficient at scale 2^{j_2}.

From equation (86) and since the scaling function ϕ is a real-valued function, the scaling coefficient $c^{j_1}[p,q]$ is expressed by

$$c^{j_1}[p,q] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} Z(u_1,u_2)\phi_{j_1p}(u_1)\phi_{j_1q}(u_2)du_1du_2. \quad (106)$$

By substituting

$$Z(u_1,u_2)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}e^{i(\omega_1u_1+\omega_2u_2)}d\dot{Z}(\omega_1,\omega_2)$$

into (106), we have

$$\begin{aligned}c^{j_1}[p,q] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}e^{i(\omega_1u_1+\omega_2u_2)}d\dot{Z}(\omega_1,\omega_2)\phi_{j_1p}(u_1)\phi_{j_1q}(u_2)du_1du_2\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(\int_{-\infty}^{\infty}e^{i\omega_1u_1}\phi_{j_1p}(u_1)du_1)(\int_{-\infty}^{\infty}e^{i\omega_2u_2}\phi_{j_1q}(u_2)du_2)d\dot{Z}(\omega_1,\omega_2)\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\overline{\widehat{\phi}_{j_1p}(\omega_1)\widehat{\phi}_{j_1q}(\omega_2)}d\dot{Z}(\omega_1,\omega_2),\end{aligned}$$

using

$$\widehat{\phi}_{j_1p}(\omega_1)=\int_{-\infty}^{\infty}e^{-i\omega_1u_1}\phi_{j_1p}(u_1)du_1,$$

and

$$\widehat{\phi}_{j_1q}(\omega_2)=\int_{-\infty}^{\infty}e^{-i\omega_2u_2}\phi_{j_1q}(u_2)du_2.$$

Similarly

$$c^{j_2}[m,n]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\overline{\widehat{\phi}_{j_2m}(\omega_1')\widehat{\phi}_{j_2n}(\omega_2')}d\dot{Z}(\omega_1',\omega_2').$$

Since $c^{j_2}[m,n]$ is real, it is equal to its complex conjugation

$$c^{j_2}[m,n]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\widehat{\phi}_{j_2m}(\omega_1')\widehat{\phi}_{j_2n}(\omega_2')\overline{d\dot{Z}(\omega_1',\omega_2')}.$$

Next, by computing the covariance of the two scaling coefficients, we have

$$\begin{aligned}&E\{c^{j_1}[p,q]c^{j_2}[m,n]\}\\ =\;& E\{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\overline{\widehat{\phi}_{j_1p}(\omega_1)\widehat{\phi}_{j_1q}(\omega_2)}d\dot{Z}(\omega_1,\omega_2)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\widehat{\phi}_{j_2m}(\omega_1')\widehat{\phi}_{j_2n}(\omega_2')\overline{d\dot{Z}(\omega_1',\omega_2')}\}.\end{aligned}$$

Using the third result of the Spectral Representation Theorem

$$E\{c^{j_1}[p,q]c^{j_2}[m,n]\}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\overline{\widehat{\phi}_{j_1p}(\omega_1)\widehat{\phi}_{j_1q}(\omega_2)}\widehat{\phi}_{j_2m}(\omega_1)\widehat{\phi}_{j_2n}(\omega_2)S(\omega_1,\omega\ )d\omega_1d\omega_2,\tag{107}$$

where

$$S(\omega_1,\omega_2)=\frac{1}{(2\pi)^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}C(h_1,h_2)e^{-i(\omega_1h_1+\omega_2h_2)}dh_1dh_2.$$

Using the properties of Fourier transform together with the definition of the function ϕ_{j_1p} we have

$$\overline{\widehat{\phi}_{j_1p}(\omega)}=2^{j_1/2}e^{i\omega 2^{j_1}p}\overline{\widehat{\phi}(2^{j_1}\omega)} \tag{108}$$

and

$$\widehat{\phi}_{j_2m}(\omega)=2^{j_2/2}e^{-i\omega 2^{j_2}m}\widehat{\phi}(2^{j_2}\omega) \tag{109}$$

By substituting (108) and (109) into equation (107) and simplifying the result, we have

$$E\{c^{j_1}[p,q]c^{j_2}[m,n]\}=K_1\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}K_2\overline{\widehat{\phi}(2^{j_1}\omega_1)\widehat{\phi}(2^{j_1}\omega_2)}\widehat{\phi}(2^{j_2}\omega_1)\widehat{\phi}(2^{j_2}\omega_2)S(\omega_1,\omega_2)d\omega_1d\omega_2, \tag{110}$$

where

$$K_1=2^{(j_1+j_2)/2},$$

and

$$K_2=e^{i\omega_1 2^{j_1}(p-2^{j_2-j_1}m)}e^{i\omega_2 2^{j_1}(q-2^{j_2-j_1}n)}.$$

It can be seen that the covariance between the two scaling coefficients $c^{j_1}[p,q]$ and $c^{j_2}[m,n]$ depends only on $(p-2^{j_2-j_1}m, q-2^{j_2-j_1}n)$. In particular, the covariance of two scaling coefficients at the same scale $c^j[p,q]$ and $c^j[m,n]$ depends on the difference of the indices $(p-m, q-n)$. ■

These properties are used to simulate a scaling coefficient at one scale from scaling coefficient(s) at other scale.

3.2.4 Wavelet Spectrum - Scaling Spectrum and Distributions of the Wavelet and Scaling Coefficients

The wavelet spectrum, denoted by ζ_k^j, and the scaling spectrum, denoted by ξ^j, are defined as the expected values of the squares of the wavelet coefficients and scaling coefficients, respectively. The wavelet spectrum and the scaling spectrum obtained by the discrete wavelet transform of a realisation of a second order stationary random function Z are independent of location (Sachs et al, 2000). If Z is a zero mean Gaussian random function, the wavelet coefficients and scaling coefficients follow zero mean normal distributions with variance $\zeta_k^j, k = 1, 2, 3$, and ξ^j, respectively that is

$$d_k^j[n,m] \sim N(0, \zeta_k^j) \tag{111}$$

and

$$c^j[n,m] \sim N(0, \xi^j). \tag{112}$$

The scaling spectrum ξ^j and the wavelet spectrum ζ_k^j can be approximated via equations (98) and (100)-(102) respectively, by replacing p with m and q with n. That is, the scaling spectrum is approximated by

$$\xi^j = \sum_{u_1,v_1=0}^{2M-1} \sum_{u_2,v_2=0}^{2M-1} h[u_1]h[v_1]h[u_2]h[v_2]\beta^{j-1}_{(2m+u_1,2m+u_2),(2n+v_1,2n+v_2)}, \tag{113}$$

and the wavelet spectra are approximated by

$$\zeta_1^j = \sum_{u_1,v_1=0}^{2M-1} \sum_{u_2,v_2=0}^{2M-1} h[u_1]g[v_1]h[u_2]g[v_2]\beta^{j-1}_{(2m+u_1,2m+u_2),(2n+v_1,2n+v_2)}, \tag{114}$$

$$\zeta_2^j = \sum_{u_1,v_1=0}^{2M-1} \sum_{u_2,v_2=0}^{2M-1} g[u_1]h[v_1]g[u_2]h[v_2]\beta^{j-1}_{(2m+u_1,2m+u_2),(2n+v_1,2n+v_2)}, \tag{115}$$

$$\zeta_3^j = \sum_{u_1,v_1=0}^{2M-1} \sum_{u_2,v_2=0}^{2M-1} g[u_1]g[v_1]g[u_2]g[v_2]\beta^{j-1}_{(2m+u_1,2m+u_2),(2n+v_1,2n+v_2)}. \tag{116}$$

3.3 Applications of the Discrete Wavelet Transform in Multilevel Wavelet based Simulation Algorithms

Equations (93) and (97) indicate that each scaling coefficient at a coarser scale is equal to twice the weighted average of scaling coefficients in an associated square window whose width is equal to the wavelet filter length. In addition, as can be seen in Figure 7, the size of the scaling image at the coarser scale is equal only to a quarter of the size of the scaling image at the finer scale, yet it preserves all features of this scaling image. Therefore, the discrete wavelet transform can be used to shrink a study region whose attribute values are known only at some locations to a region whose attribute values at all locations are known without distorting the spatial structure of the original region.

Equations (98)-(105) allow unknown scaling or wavelet coefficients to be computed from the known ones. The correlation between pairs of wavelet and scaling coefficients at the same coarser scale makes it possible to deduce the wavelet coefficients from the scaling coefficients at the same scale. Having both scaling and wavelet coefficients at a coarser scale, the scaling image at the previous finer scale can be suitably reconstructed from the scaling image at the coarser scale using the inverse discrete wavelet transform. This property is used in the algorithm PWSIM introduced in Chapter 7.1.

The correlation between pairs of scaling coefficients at two consecutive scales can be used to compute the scaling coefficients at the finer scale from those at the coarser scale and vice versa. Therefore, the simulation of the scaling image at the original scale from the scaling image at the coarser scales can be carried out using the scaling coefficients alone. This property and the method of shrinking the study region using the discrete wavelet transform are used in the algorithm DWSIM introduced in

Chapter 7.2.

Scaling and wavelet spectra together with their properties are used in the single-level wavelet-based simulation algorithms HSIM and DB2SIM in Chapter 6.

4 PERFORMANCE & EVALUATION MEASURES

In this Chapter we describe the criteria used to assess the performance of the simulation algorithms in this thesis. The criteria used are both quantitative and visual. Quantitative assessment consists of measures for local accuracy and global accuracy; visual assessment is carried out via the mosaic plots of the simulated realisations. For the assessment of the local accuracy of the results, we use misclassification analysis, E-type estimates and the absolute errors between the E-type estimates and the true values and conditional variances, computed for the attribute values. In addition, the *accuracy*, *precision* and *goodness* (Deutsch, 1997), computed in the normal score space, of the results obtained via simulation are also considered. For the assessment of the global accuracy, we focus on the reproduction of the sample histogram and the semivariogram model computed in the normal score space.

4.1 Misclassification Analysis

Misclassification analysis can be used as a measure of the local accuracy of a stochastic or deterministic model. For example, in soil science applications, each location in the study region can be classified into two categories: a location $\mathbf{u}$ is safe from some type of pollution if the value at that location lies below a predefined tolerance z_c, and a location is unsafe if the value lies above the tolerance. That is

$$I(\mathbf{u}) = \begin{cases} 0 & if \quad z(\mathbf{u}) \leq z_c \\ 1 & if \quad z(\mathbf{u}) > z_c \end{cases}$$

where the value 0 denotes a safe location and 1 denotes an unsafe location.

In comparing the category at each location obtained from a simulated realisation with the actual one, four cases occur: a safe location is correctly classified, an unsafe

location is correctly classified, a safe location is wrongly classified as unsafe and an unsafe location is wrongly classified as safe. The probability that a safe location u is misclassified as unsafe is given by

$$\alpha(\mathbf{u}) = Prob\{Z(\mathbf{u}) \leq z_c | z_L^*(\mathbf{u}) > z_c, (n)\} \tag{117}$$

where $z_L^*(\mathbf{u})$ denotes the simulated value at location u. The probability that an unsafe location u is misclassified as safe is given by

$$\begin{aligned} \beta(\mathbf{u}) &= Prob\{Z(\mathbf{u}) > z_c | z_L^*(\mathbf{u}) \leq z_c, (n)\} \\ &= 1 - \alpha(\mathbf{u}). \end{aligned} \tag{118}$$

At a given location either (117) or (118) is defined, but not both.

4.2 E-type Estimate and Absolute Error

Another measure of local accuracy used in this thesis is the E-type estimate and the absolute average error computed at each unsampled location. The E-type estimate at location u is the average of L simulated values at this location obtained by running a simulation algorithm L times. The absolute average error $AAE(\mathbf{u})$ at location u is the absolute difference between the true value $z(\mathbf{u})$ and the E-type estimate at location u. This measure is computed by

$$AAE(\mathbf{u}) = \left| z(\mathbf{u}) - \frac{1}{L} \sum_{l=1}^{L} z^l(\mathbf{u}) \right|$$

where $z^l(\mathbf{u}), l = 1, 2, ...L$ are simulated values at u.

4.3 Conditional Variance

Let L be the number of simulated realisations obtained by running a simulation algorithm L times, where L is large. At each location $\mathbf{u}_i$ we obtain a set of L condi-

tional simulated values $\{z^l(\mathbf{u}_i), l = 1, 2, ..., L\}$ that define a local frequency distribution. The conditional variance at location $\mathbf{u}_i$ is the variance of the simulated values $\{z^l(\mathbf{u}_i), l = 1, 2, ..., L\}$.

4.4 Sample Histogram Reproduction

A quantitative measure fo. sample histogram reproduction is the mean of the absolute deviation (MAD) between the sample quantiles and the quantiles of an individual realisation. This measure is computed by

$$MAD = \frac{1}{n}\sum_{i=1}^{n}|z_i^s - z_i| \tag{119}$$

where z_i^s and z_i are the quantiles for the realisation and the sample in the normal score space, respectively and n is the number of quantiles.

4.5 Semivariogram Reproduction

For semivariogram reproduction we use the mean of the absolute relative errors (MAE) between the directional experimental semivariogram of the normal scores of an individual realisation and the semivariogram model fitted to the normal scores of the associated sample. The formula for MAE in the isotropic case is

$$MAE = \frac{1}{n_l}\sum_{i=1}^{n_l}\left|\frac{g^s(|\mathbf{h}_i|) - g(|\mathbf{h}_i|)}{g(|\mathbf{h}_i|)}\right| \tag{120}$$

where g^s denotes the experimental semivariogram for the realisation, g denotes the semivariogram model, n_l is the number of lags used to compute the errors and $|\mathbf{h}_i|$ is the distance satisfying $0 < |\mathbf{h}_i| \leq range$. Note that in equation (120), the value associated with $|\mathbf{h}_i| = 0$ is excluded for the case of a semivariogram model that has a nugget effect component. The sum is computed only up to the range since only

the spatial continuity and the errors at distances smaller or equal to the range are regarded as important. The division by the value of the semivariogram model will give more weight to the spatial continuity at short distances.

In the anisotropic case, for simplification, the measure for semivariogram model reproduction is based on the average of MAE values computed in the directions of maximum and minimum spatial continuity.

4.6 Accuracy - Goodness - Precision

Let us consider a set of L simulated realisations obtained by running a conditional simulation algorithm L times. The results obtained by the application of a transfer function (for example a flow simulator) to each of these L realisations model a probability distribution. This outcome probability distribution is *accurate* if the true result obtained by applying the same transfer function to the true values lies within some predefined symmetric probability interval of this probability distribution. The *precision* of the outcome distribution is defined by its spread; the more spread the less precise. The accuracy of the outcome distribution can only be evaluated based on the known values. These known values can be the sample data used in the modelling, the data held back before the modelling (jackknife data) or some additional information about the phenomenon.

The adequacy of the algorithm can also be assessed directly using symmetric coverage probabilities as outlined by Deutsch (1997). According to this method a stochastic model is *accurate* if $\overline{\xi(p)}$, the proportion of the true values lying in an interval $(p_{low}, p_{up}]$ of length p, is greater than or equal to p, where

$$p_{low} = \frac{1-p}{2} \text{ and } p_{up} = \frac{1+p}{2}$$

and $p \in [0,1]$. It can be seen that $(p_{low}, p_{up}]$ is symmetric about the value 0.5. The *goodness* of the outcome distribution is determined by the closeness of the value $\overline{\xi(p)}$ to p.

To save computational time, in this thesis the true values used to compute the accuracy and goodness of a stochastic model are the jackknife data, which will be described in Chapter 5. Based on this jackknife approach the computation of $\overline{\xi(p)}$ is carried out as follows.

Consider a sample of n values taken at locations $\mathbf{u}_\alpha, \alpha = 1, ..., n$ denoted by

$$\{z(\mathbf{u}_\alpha), \alpha = 1, ..., n\} \tag{121}$$

which is used in the modelling and conditioning of realisations of a random function Z in some study region A. Let $\mathbf{u}_i, i = 1, 2, ..., Q$, where Q is the number of jackknife data, be the locations where the known values are held back. By running a simulation algorithm L times, where L is large, at each location $\mathbf{u}_i$ we obtain a set of L conditional simulated values $\{z^l(\mathbf{u}_i), l = 1, 2, ..., L\}$ which define a local probability distribution.

If $\widehat{F}(\mathbf{u}_i; z|(n))$ is the conditional cumulative probability distribution function at location $\mathbf{u}_i$, defined by

$$\widehat{F}(\mathbf{u}_i; z|(n)) = Prob\{Z(\mathbf{u}_i) \leq z|(n)\}$$

the conditional cumulative probability for the true value $z(\mathbf{u}_i)$ at location $\mathbf{u}_i$ is

$$\widehat{F}(\mathbf{u}_i; z(\mathbf{u}_i)|(n)) = Prob\{Z(\mathbf{u}_i) \leq z(\mathbf{u}_i)|(n)\}.$$

The proportion $\overline{\xi(p)}$ of the true values that lie within the probability interval for all unsampled locations then is computed by

$$\overline{\xi(p)} = \frac{1}{Q} \sum_{i=1}^{Q} \xi(\mathbf{u}_i, p). \tag{122}$$

where

$$\xi(u_i,p) = \begin{cases} 1 & \text{if } \widehat{F}(u_i; z(u_i)|(n)) \in (p_{low}, p_{up}] \\ 0 & \text{otherwise} \end{cases} . \tag{123}$$

Definition 6 *A stochastic model is said to be accurate if* $\overline{\xi(p)} \geq p$ *for all* p.

The *accuracy* of a stochastic model can be visualised by the plot of $\overline{\xi(p)}$ against p and a stochastic model is accurate if all of the points $(p, \overline{\xi(p)})$ lie above or on the line that bisects the first quadrant (the 45° line). This plot is called the *accuracy plot*.

In order to assess the performance of a simulation algorithm, two measures are defined, one for the *accuracy* and one for the *goodness*. The measure for the accuracy is computed by

$$A = \int_0^1 a(p)dp \tag{124}$$

where

$$a(p) = \begin{cases} 1 & \text{if } \overline{\xi(p)} \geq p \\ 0 & \textit{otherwise} \end{cases}$$

for $p \in (0, 1]$. If all the pairs $(p, \overline{\xi(p)})$ in the accuracy plot lie above or on the line that bisects the first quadrant, that is $\overline{\xi(p)} \geq p$ for all p, then the coefficient A attains its maximum value of 1. If all the pairs $(p, \overline{\xi(p)})$ lie below that line, that is $\overline{\xi(p)} < p$ for all p, then A attains its minimum value of 0. In practice, to compute (124) the interval $(0, 1]$ is partitioned into K evenly-spaced intervals of length $1/K$ and the value A is obtained numerically by the discrete sum

$$A = \frac{1}{K}\sum_{k=1}^{K} a(p_k)$$

where $p_k = k/K$.

The goodness measure is computed by

$$G = 1 - \int_0^1 [3a(p) - 2]|\overline{\xi(p)} - p|dp. \tag{125}$$

Similar to the case of the accuracy coefficient A, in practice, the computation of G is carried out numerically via discrete summations.

According to equation (125), if all the points $(p, \overline{\xi(p)})$ lie on the 45° line, that is $\overline{\xi(p)} = p$, the value G takes its maximum value of 1. The coefficient G gets smaller as the departure of the pairs $(p, \overline{\xi(p)})$ from the 45° line increases. In addition, the penalty is weighted twice for the case $\overline{\xi(p)} < p$. The goodness coefficient G can be used to deduce the location of the result obtained by applying a transfer function to the true values in relation with the median of the probability distribution modelled from the values obtained by applying the same transfer function to the realisations. The closer the value G is to 1, the closer is the median to the true result. If G is too small, the true result may fall outside the interquartiles or even outside the range of the results obtained by the simulation. This will lead to a wrong prediction.

The *precision* of a stochastic model can be determined via the average of the local probability distribution variances denoted by $\overline{V}$. The larger the average variance $\overline{V}$, the less precise is the result obtained by the simulation. Let $\sigma^2(\mathbf{u}_i)$ be the conditional variance (defined in section 4.3) of the simulated values at location $\mathbf{u}_i$ and Q be the number of jackknife data, the average variance denoted by $\overline{V}$ is computed by

$$\overline{V} = \frac{1}{Q}\sum_{i=1}^{Q} \sigma^2(\mathbf{u}_i) \tag{126}$$

Since the average variance $\overline{V}$ only provides the global spread of the conditional cummulative distribution, another measure of the precision based on the width of the local probability intervals has been introduced. This measure is the average width of the probability intervals that contain the true values and is denoted by $\overline{W}(p)$. For a

given probability p, the average width $\overline{W}$ is computed by (Goovaerts,2003)

$$\overline{W}(p) = \frac{1}{Q\bar{\xi}(p)} \sum_{i=1}^{Q} (\xi(u_i;p)[\widehat{F}^{-1}(u_i;(1+p)/2) - \widehat{F}^{-1}(u_i;(1-p)/2)])$$

where $\widehat{F}^{-1}$ denotes the inverse of the conditional cummulative probability function at location u_i.

4.7 Mosaic Plots

The performance of the algorithms is assessed visually by comparing a mosaic plot of the simulated realisation with the corresponding post plot of the associated sample and the corresponding mosaic plot of the exhaustive data set (since here the exhaustive data set is available). By observing the mosaic plot of a simulated realisation, one can detect any unusual behaviour, for example artefacts, or can verify that some particular feature seen in the post plot of the sample and / or in the mosaic plot of the exhaustive data set is also found in the simulated realisation. The mosaic plot of a realisation can also reveal whether or not the spatial continuity is reproduced.

4.8 Applications and Case Studies

The measures discussed in this Chapter will be used to evaluate the performance of the algorithms developed in this thesis through their applications to samples that have different spatial structures and that have been obtained using different sampling methods. In the case when the underlying function is multivariate normal, these algorithms are applied to seven samples with different sample sizes and sampling methods drawn from the normal scores of two exhaustive data sets, one isotropic and one anisotropic, as illustrated in the case studies in Chapters 6, 7, 8 and 9. In order to identify artefacts and the reproduction of global statistics, the performance

measures used in this case are the mosaic plots of typical realisations, the measures for the reproduction of the sample histogram and semivariogram model (the MAD and MAE values). Since the results in this case are in normal scores, the assessment of the local accuracy and the variability of the simulation is based on the accuracy, goodness and precision coefficients.

In the case when the underlying random function is not necessarily multivariate standard normal, the algorithms will be applied to two samples drawn from the attribute values of the above data sets, as illustrated in Chapter 10. In this case, the samples have to be transformed into the normal scores before the application of the algorithms. The realisations then have to be back-transformed into the attribute values after the conclusion of the algorithms. The performance measures to identify the artefacts and the reproduction of global statistics are also the mosaic plots of typical realisations and the MAD and MAE values. Since the results now are the attribute values, with the presence of the exhaustive data sets, the local accuracy is assessed via misclassification analysis and E-type estimates. The variability of the simulation is assessed via conditional variances

5 DATA SETS AND SAMPLES

The performances of the simulation algorithms in this thesis are assessed using nine samples and seven sets of jackknife data. In the case where the underlying random function is multivariate standard normal, the main samples used for the performance assessment in Chapters 6, 7 and 8 are *pH100n* and *pH100IRn* in the isotropic case and *B64n* and *B64IRn* in the anisotropic case. In addition, the samples *pH50IRn*, *pH100IRn*, *pH250IRn* and *pH500IRn* will be used to assess the impact of the sample size in Chapter 9 and the jackknife data *JackpH100n*, *JackpH100IRn*, *JackB64n*, *JackB64IRn*, *JackpH50n*, *JackpH250n* and *JackpH500n* are used in Chapters 8 and 9 for accuracy assessment. The above 14 samples are taken from the normal transformation of two exhaustive data sets; one is isotropic and the other is anisotropic. In the case where the random function is not multivariate normal, the samples used in the performance assessment in Chapter 10 for the isotropic and anisotropic cases are *pHsamp* and *Bersamp* respectively. These two samples are drawn from the attribute values of the two exhaustive data sets discussed above. A detailed description of the data sets and the samples is given below.

5.1 The Data Set pHsoil

The exhaustive data set *pHsoil* contains 3721 pH values on a 61×61 regular grid. It has been obtained by sequential Gaussian simulation of soil samples taken originally in an uncropped field in an investigation of soil salinity and acidity in the Jimperding Brook catchment area in Western Australia (Bloom and Kentwell, 1999). Summary statistics and the histogram of this data set in Figure 8A, together with the normal probability plot in Figure 8B, indicate that the data set *pHsoil* is not standard normal.

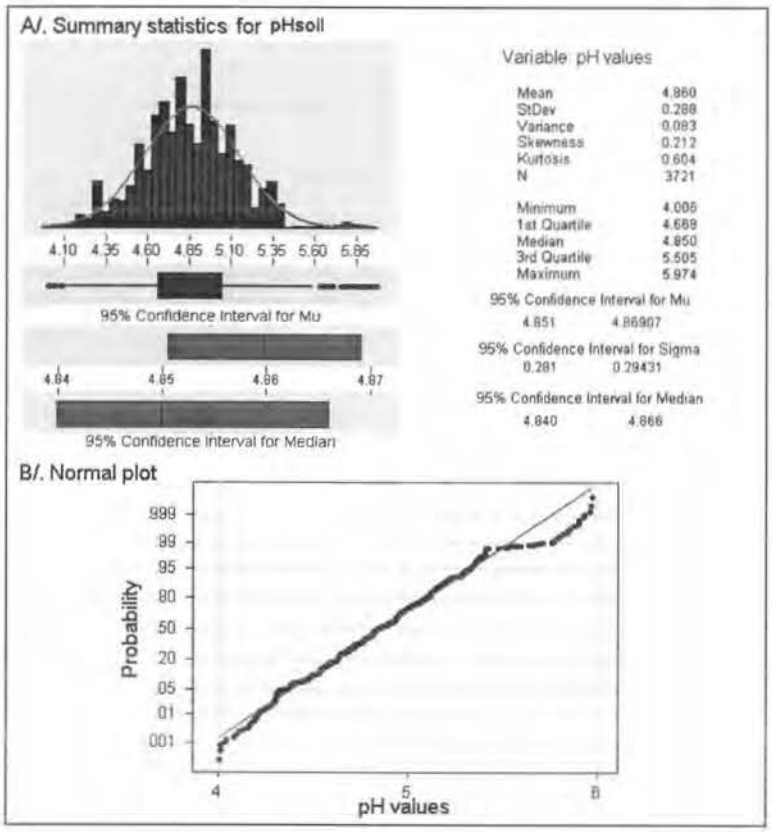


Figure 8: Summary statistics and normal probability plot for *pHsoil*.

Since *pHsoil* is not standard normal and the simulation algorithms investigated in this thesis require that the underlying random function be multivariate standard normal, the pH values have been transformed into the normal score space and the data set that contains these standard normal scores is denoted by *pHsoiln*. The mosaic plot of *pHsoil* using the left legend (or of *pHsoiln* using the right legend) and the variogram surface of *pHsoiln* in Figure 9 show that the spatial continuity is slightly stronger in the direction of azimuth 135^0. To avoid the influence of the choice of the parameters in the backtransformation on the result of the algorithms, the data set *pHsoiln* together with its associated samples, described later in this Chapter, will be used to test the performance of the algorithms presented in this thesis in the case when the underlying random function is multivariate standard normal. The data set

pHsoil and an associated sample are used as test data in the case when the underlying random function is not necessarily multivariate standard normal.

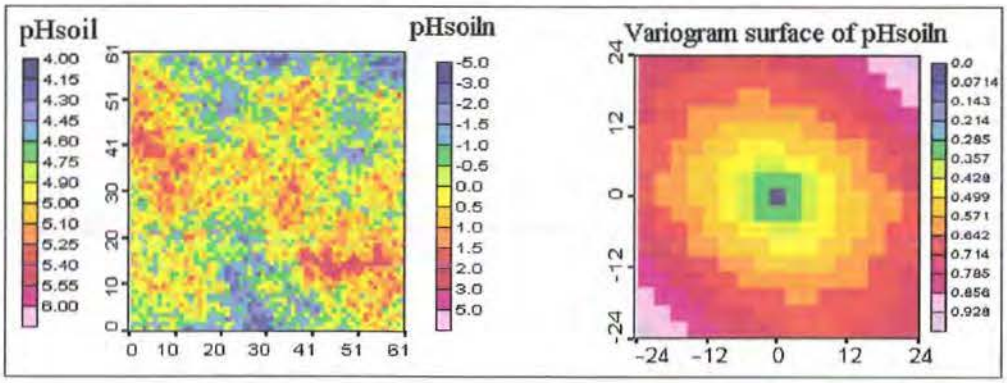


Figure 9: Mosaic plot of *pHsoil* (*pHsoiln*) and variogram surface of *pHsoiln*.

To test the performance of the simulation algorithms in the case when the random function is multivariate standard normal, we use one gridded sample, denoted by *pH100n*, and four nested ungridded samples, denoted by *pH50IRn*, *pH100IRn*, *pH250IRn* and *pH500IRn*, from *pHsoiln*. Samples *pH100n* and *pH100IRn* are used in Chapters 6, 7 and 8 to test the performance of the simulation algorithms with regard to the sampling method. Samples *pH500IRn*, *pH250IRn*, *pH100IRn* and *pH50IRn* are used in Chapter 9 to assess the impact of the sample size. In Chapter 10, to test the performance of the algorithms in practice, we use sample *pHsamp* which was randomly drawn from the exhaustive data set *pHsoil*.

5.1.1 Sample pH100n

The gridded sample *pH100n* consists of 100 values at 6^{th} node of *pHsoiln* starting from $x = 2$, $y = 2$ with grid spacing 6 units of measurement. Summary statistics for *pH100n* are shown in Figures 10.

The post plot and the variogram surface for *pH100n* in Figures 11A and 11B show

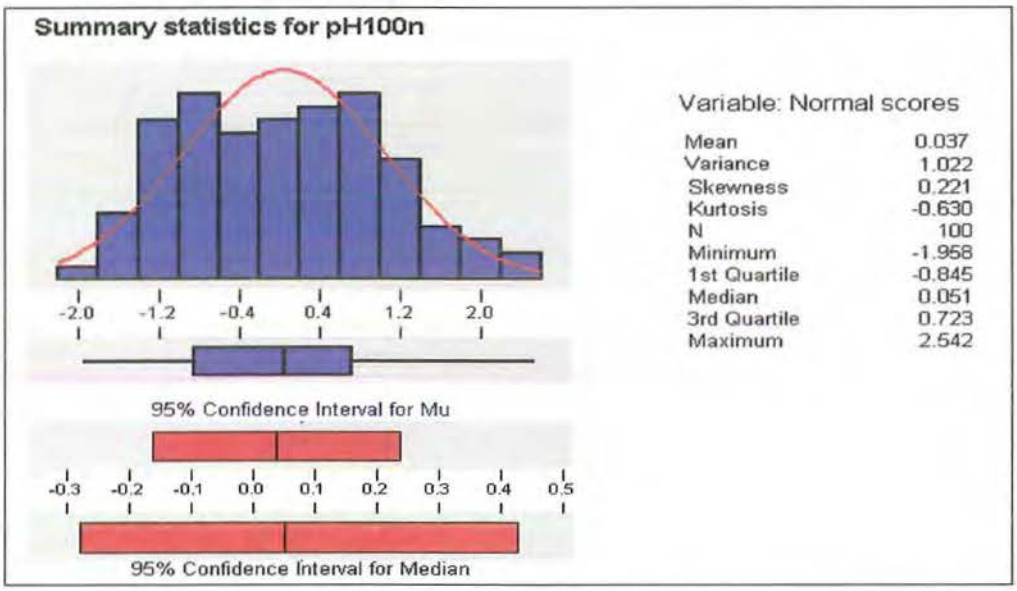


Figure 10: Summary statistics for *pH100n*.

that *pH100n* is an isotropic sample, with both maps indicating that spatial continuity is the same in all directions. As *pH100n* is isotropic, only the omnidirectional experimental semivariogram is modelled and the standard exponential semivariogram model we have fitted to the experimental semivariogram with lag spacing $|\mathbf{h}| = 6.0$ is shown in Figure 11C. This semivariogram model has nugget 0.15, sill 0.85 and range 18.00.

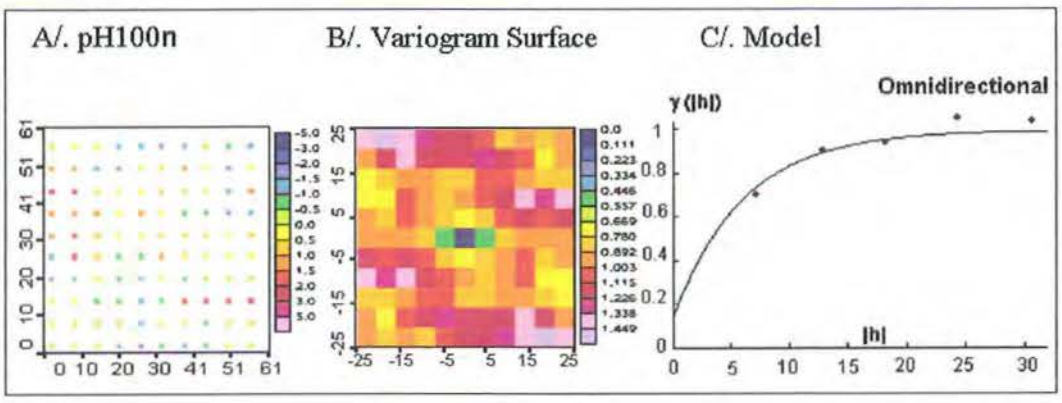


Figure 11: Post plot of *pH100n* and its variography.

5.1.2 The Ungridded Samples

The four ungridded samples from *pHsoiln* are nested. Sample *pH500IRn* consists of 500 values randomly drawn from the data set *pHsoiln;* sample *pH250IRn* consists of 250 values randomly drawn from *pH500IRn;* sample *pH100IRn* consists of 100 values randomly drawn from *pH250IRn* and sample *pH50IRn* consists of 50 values randomly drawn from *pH100IRn*. In summary, the four ungridded samples satisfy

$$pH50IRn \subset pH100IRn \subset pH250IRn \subset pH500IRn \subset pHsoiln.$$

Summary statistics and the histograms for the nested samples are shown in Figure 12.

The post plots, variogram surfaces and semivariogram models for the four nested samples are shown in Figure 13. The variogram surfaces of the nested samples in the middle column indicate that all four samples are isotropic since the spatial continuity is the same in all directions. The omnidirectional experimental semivariogram, computed using lag spacing $|\mathbf{h}| = 4.0$, for sample *pH50IRn* and its associated model are shown on the right of row 1 in Figure 13; the omnidirectional experimental semivariograms for the other three samples, computed using lag spacing $|\mathbf{h}| = 1.5$, and their associated models are shown on the right of rows 2, 3 and 4. The standardised exponential semivariogram models for all four nested samples have the same parameters with nugget 0.15, sill 0.85 and range 18.00.

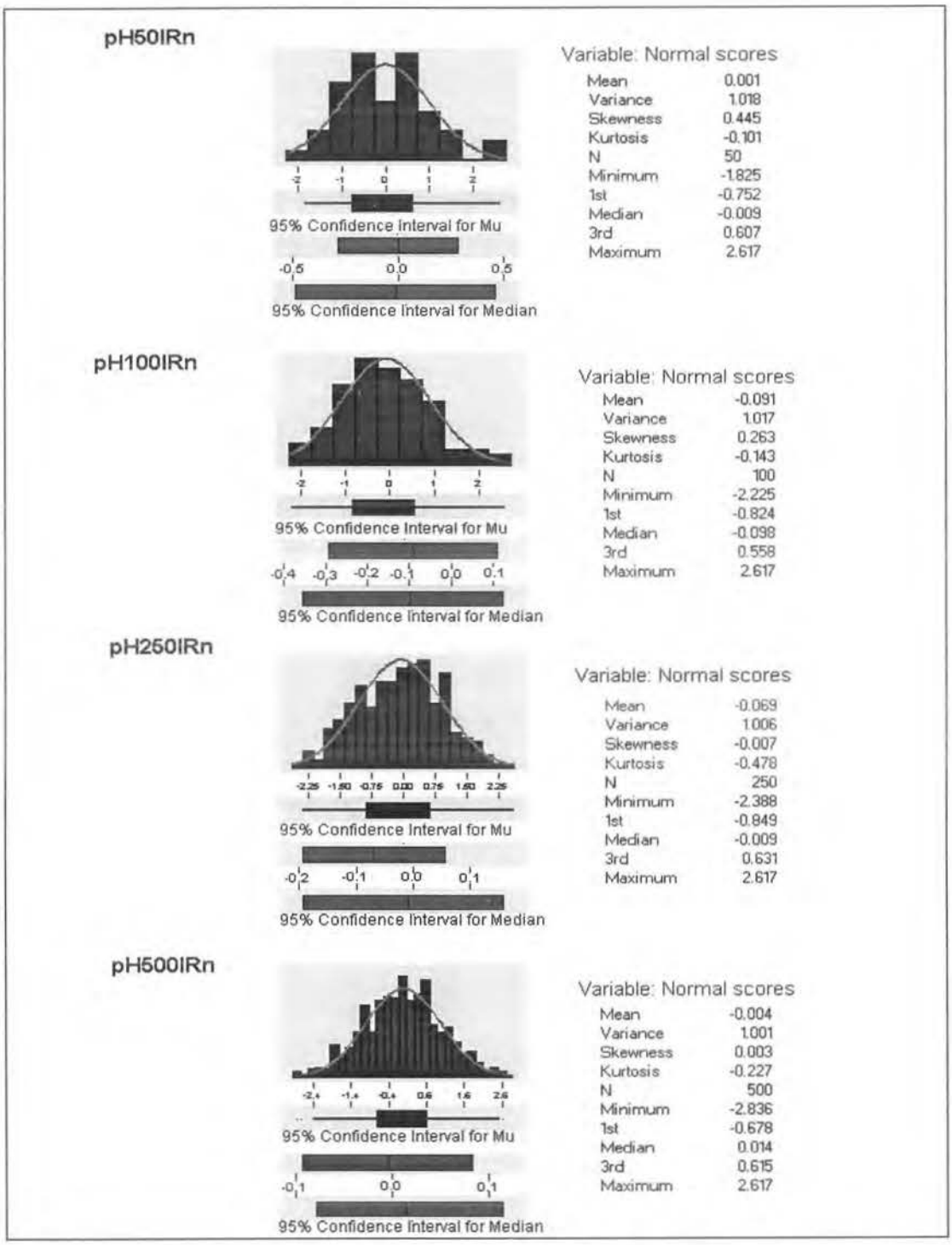


Figure 12: Summary statistics for nested samples.

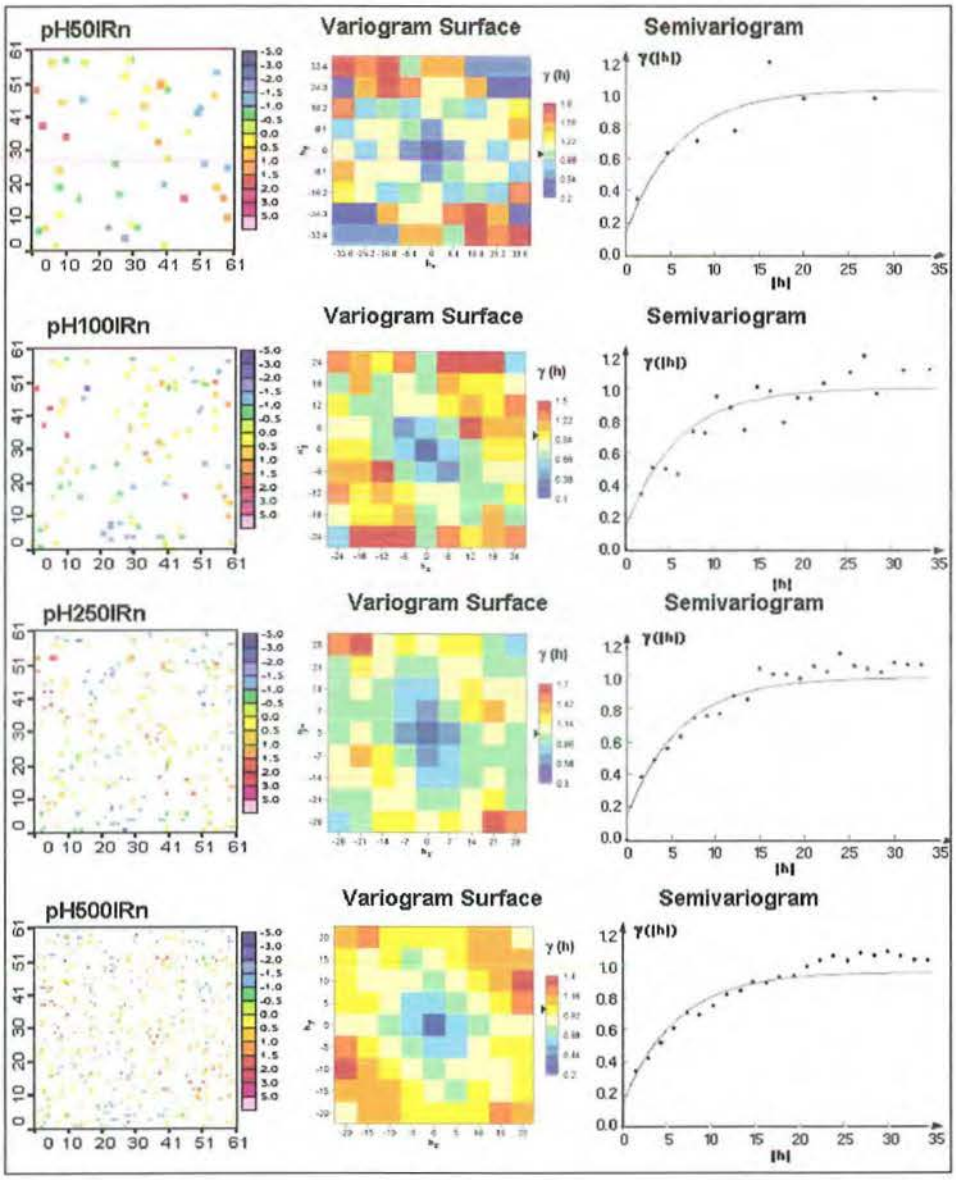


Figure 13: Post plots and the variography of ungridded samples from *pHsoiln*.

5.1.3 Jackknife Data for the Samples of pHsoiln

Associated with the samples described above are five sets of jackknife data. The jackknife data set for each sample consists of all values of the data set *pHsoiln* excluding the sample values. These jackknife data sets will be used only in Chapters 8 and 9 in the performance assessment based on the accuracy plots described in Chapter 4. Summary statistics for the jackknife data are shown in Table 2.

Statistics	jack-pH100n	jack-pH50IRn	jack-pH100IRn	jack-pH250IRn	jack-pH500IRn
N	3621	3671	3621	3471	3221
Mean	-0.001	0.000	0.003	0.005	0.001
Variance	1.000	1.000	1.000	1.000	1.000
Min	-3.644	-3.644	-3.644	-3.644	-3.644
Q1	-0.670	-0.674	-0.673	-0.667	-0.675
Median	0.001	-0.001	0.003	0.001	-0.002
Q3	0.672	0.676	0.678	0.678	0.681
Max	3.644	3.644	3.644	3.644	3.644

Table 2: Summary statistics for jackknife data of samples from *pHsoiln*.

5.1.4 Sample pHsamp

Sample *pHsamp* consists of 100 pH values randomly drawn from the exhaustive data set *pHsoil*. It is used in Chapter 10 to test the practical usefulness of the algorithm DWSIM compared with the sequential Gaussian simulation algorithm (SGSIM). Summary statistics and the cumulative probability distribution function for sample *pHsamp* are shown in Figure 14.

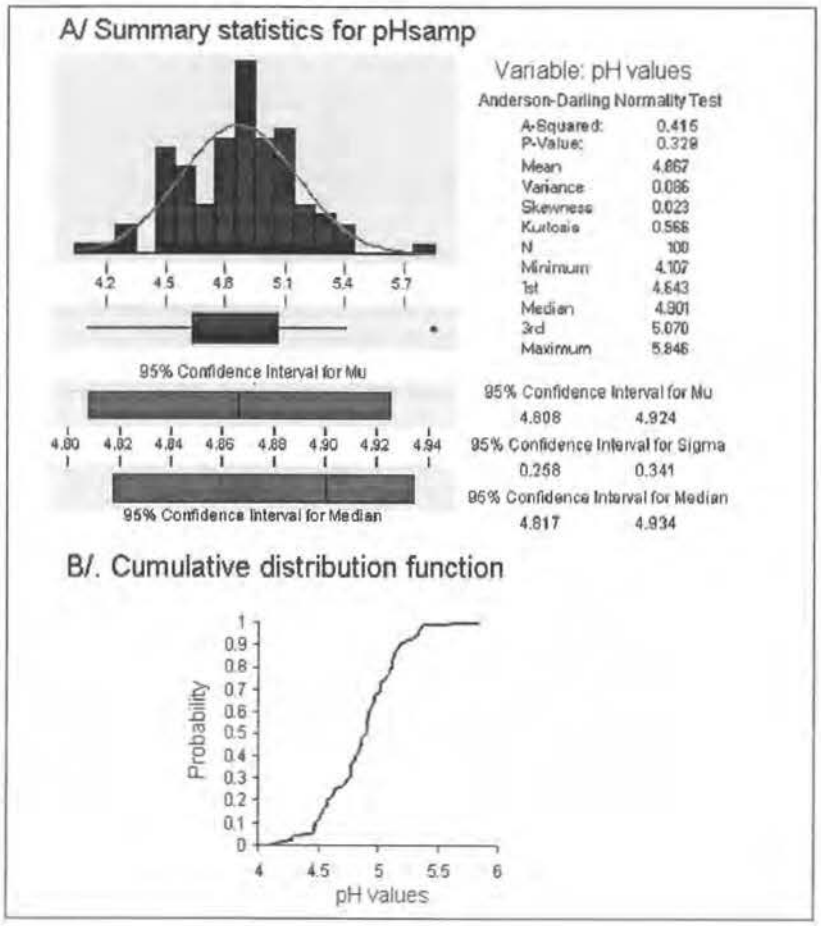


Figure 14: Summary statistics and cumulative distribution function of *pHsamp*.

Since *pHsamp* is not standard normal and since the algorithms used in this thesis are based on the assumption that the underlying random function is multivariate standard normal, the pH values, shown in the post plot on the left of Figure 15, were transformed into normal scores before the computation of the variography. The isotropic exponential semivariogram model fitted to the omnidirectional experimental semivariogram for the normal scores, computed using lag spacing 3.5, is shown on the right of Figure 15. It has nugget 0.15, sill 0.85 and range 18.00.

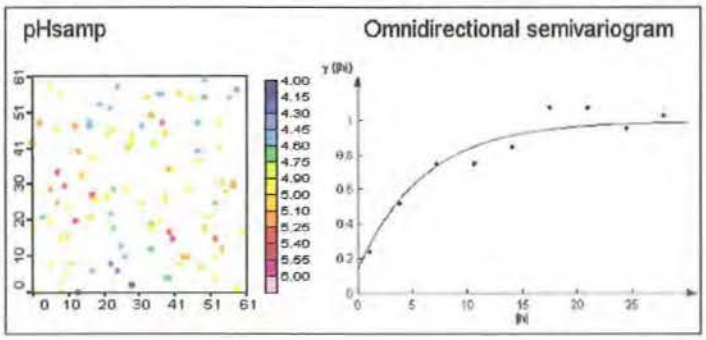


Figure 15: Post plot of *pHsamp* and semivariogram model fitted to experimental semivariogram for normal scores.

5.2 The Data Set Berea

Berea is a real two-dimensional exhaustive data set. It contains 1600 permeability measurements (in millidarcies) on a 40×40 regular grid. These measurements were taken from a 2 foot $\times 2$ foot vertical slab of Berea sandstone (Giordano et al, 1985) using an air permeameter. The permeability measurements in *Berea* do not follow a normal distribution. This can be seen from the summary statistics and the normal probability plot in Figures 16A and 16B, respectively.

As in the isotropic case, the permeability measurements are transformed into the normal score space and this data set is denoted by *Berean*. The mosaic plot of *Berea* if the legend on the left is used (or of *Berean* if the legend on the right is used) and the variogram surface of *Berean* in Figure 17 show that the spatial continuity in the direction $N55^0W$ is stronger than the spatial continuity in any other direction. It means that the data set *Berea* (or *Berean*) is anisotropic with the maximum spatial continuity at the direction $N55^0W$. Because of this spatial structure, *Berean* is used in Chapters 6, 7 and 8 and *Berea* is used in Chapter 10 to test the performance of our simulation algorithms in the anisotropic case. Performance

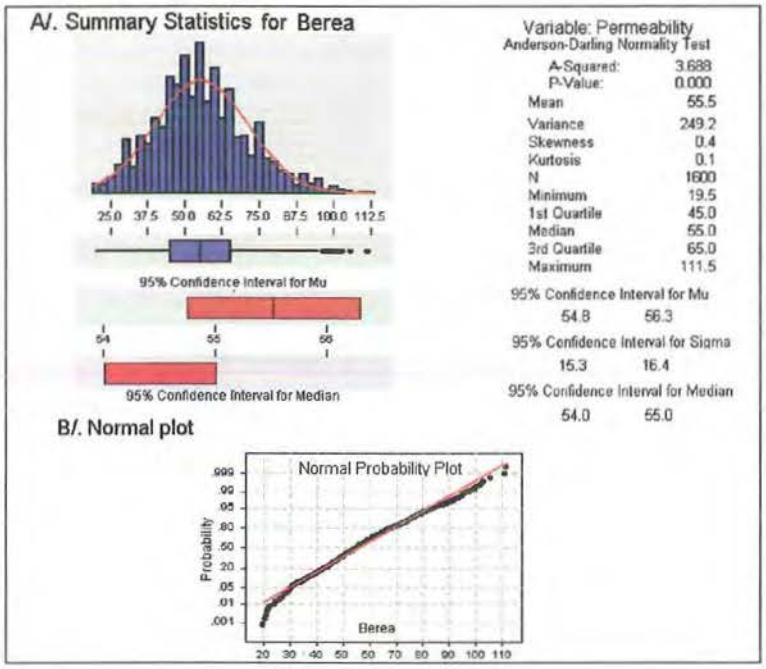


Figure 16: Summary statistics for *Berea* and normal probability plot.

assessment of the algorithm DWSIM, which will be discussed in Chapters 7.2, for highly skewed anisotropic samples can be found in Tran et al, 2002a.

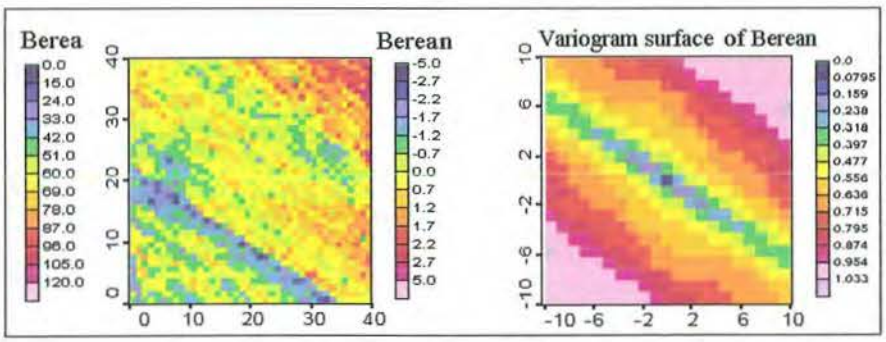


Figure 17: Mosaic plot of *Berea* (*Berean*) and variogram surface of *Berean*.

5.2.1 Samples B64n and B64IRn

To test the performance of the algorithms with regard to anisotropic data in the case where the underlying random function is multivariate standard normal in Chapters 6, 7, and 8, two samples, each of which contains 64 values, were drawn from *Berean*. In

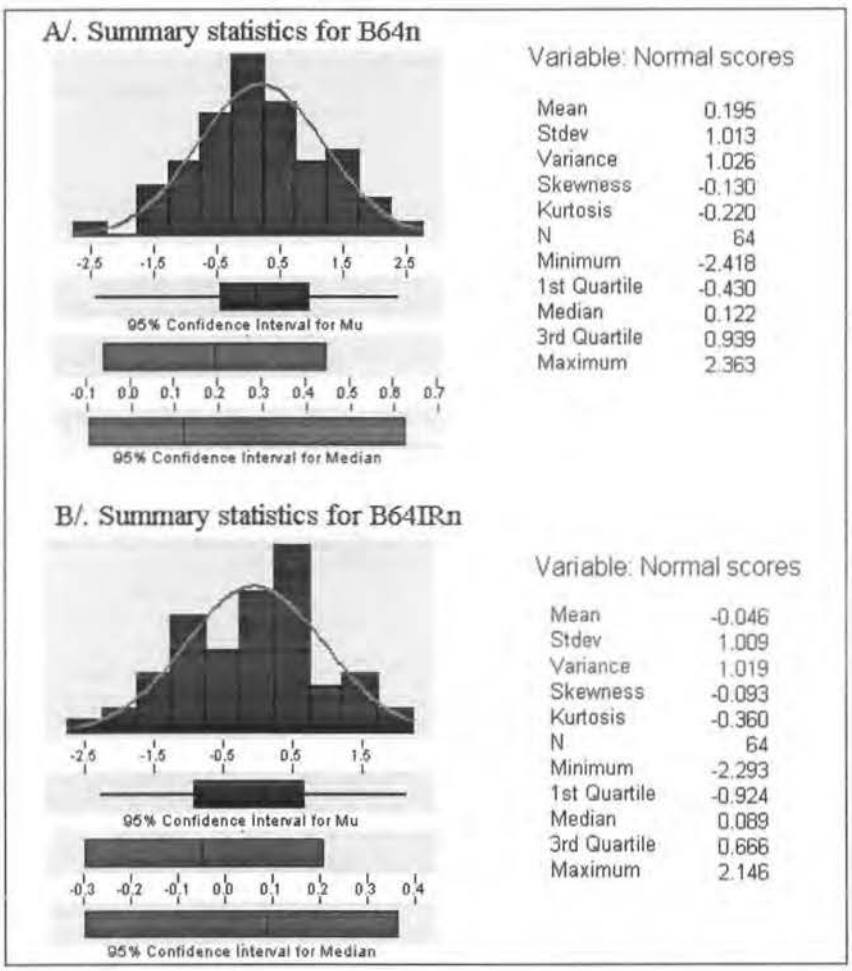


Figure 18: Summary statistics for *B64n* and *B64IRn*.

order to investigate the impact of the sampling method, one sample is gridded and the other is ungridded. These samples are denoted by *B64n* and *B64IRn*, respectively. The values of the gridded sample *B64n* have been selected on every 5^{th} node of *Berean* while those of *B64IRn* are randomly drawn from this data set. Summary statistics

for the two samples are shown in Figure 18A and Figure 18B respectively.

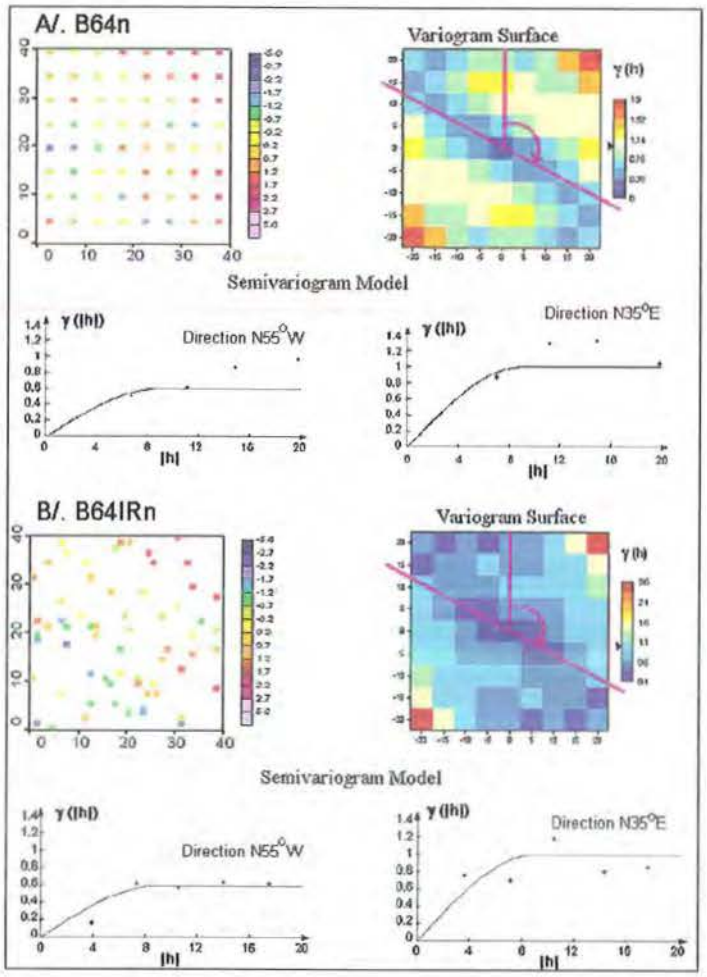


Figure 19: Post plots of *B64n*, *B64IRn* and variography.

The variogram surfaces in Figure 19 show that the spatial continuity in the direction $N55°W$ is much stronger than the spatial continuity in other directions. In addition, the experimental semivariograms for *B64n*, with lag spacing $|\mathbf{h}| = 5.0$, and for *B64IRn*, with lag spacing $|\mathbf{h}| = 3.0$, reveal that the sill values in the direction of maximum spatial continuity are lower than that in the direction of minimum spatial continuity $N35°E$. Therefore, the semivariogram model in this case is the sum of an isotropic model and a zonal component in the direction $N35°E$, as discussed in

Chapter 2. The same standardised spherical semivariogram model is used for both samples. This model has nugget, sill and range of 0.0, 0.6, and 9.0 in the direction of maximum spatial continuity ($N55°W$) and nugget, sill and range of 0.0, 1.0 and 9.0 in the direction of minimum spatial continuity ($N35°E$). This model is shown in Figures 19A and 19B.

5.2.2 Jackknife Data for the Samples of Berean

Similar to the case of isotropic samples, the jackknife data used in Chapter 8 in the performance assessment via the accuracy plot of each sample are the values of the data set *Berean* excluding the sample values. For samples *B64n* and *B64IRn*, summary statistics for the jackknife data are shown in Table 3.

Statistics	jackB64n	jackB64IRn
N	1536	1536
Mean	-0.008	0.002
Variance	0.998	1.000
Min	-3.421	-3.421
Q1	-0.683	-0.669
Median	-0.011	-0.005
Q3	0.663	0.675
Max	3.421	3.421

Table 3: Summary statistics for jackknife data of *B64n* and *B64IRn*.

5.2.3 Sample Bersamp

In order to compare the performance in practice of the algorithm DWSIM, described in Chapter 7, with the performance of the sequential Gaussian simulation algorithm (SGSIM) in the anisotropic case in Chapter 10, a random sample of 64 permeability values denoted by *Bersamp* was drawn from the exhaustive data set *Berea*. Summary statistics, together with the cumulative probability distribution function for *Bersamp*, are shown in Figure 20.

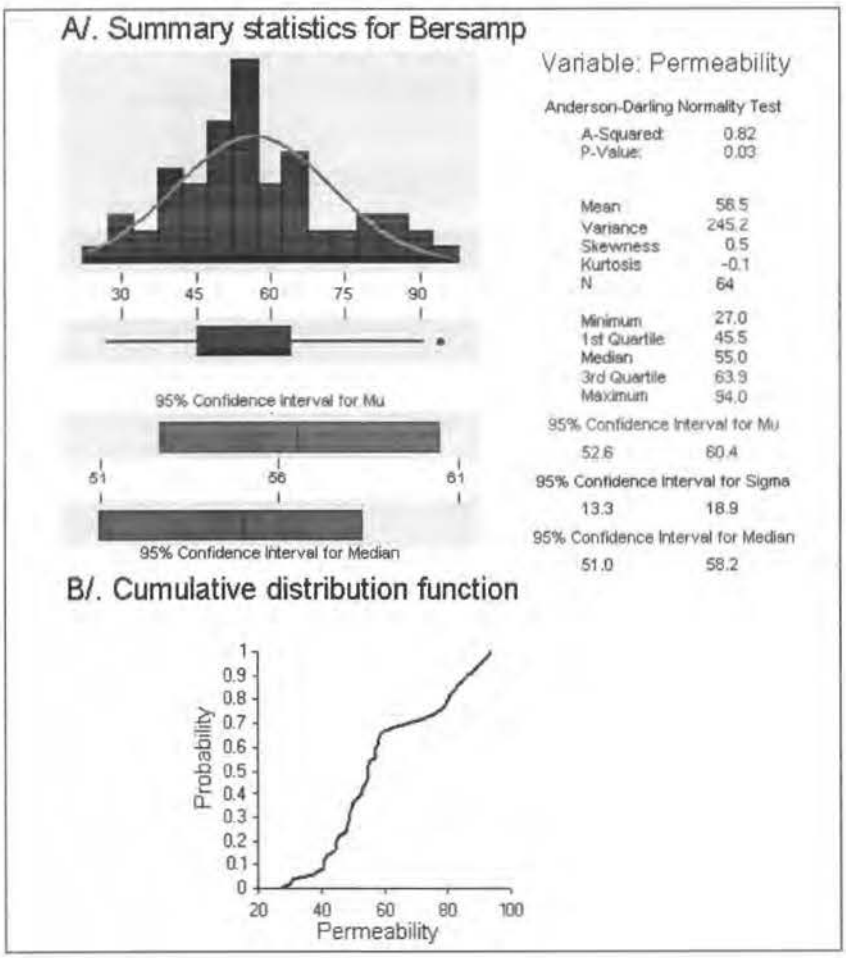


Figure 20: Summary statistics and cumulative probability plot of *Bersamp*.

Since the permeability measurements do not follow a standard normal distribu-

tion, these values were transformed into the normal score space. The post plot of the permeability values and the semivariogram models fitted to the experimental semivariograms for the normal scores in the direction of maximum and minimum spatial continuity, computed using lag spacing 3.5, are shown in Figure 21. The nugget, sill and range of these standardised spherical semivariogram models in the direction of maximum spatial continuity $N55^oW$ and in the direction of minimum spatial continuity $N35^oE$ are 0.0, 0.6, 9.0 and 0.0, 1.0 and 9.0 respectively.

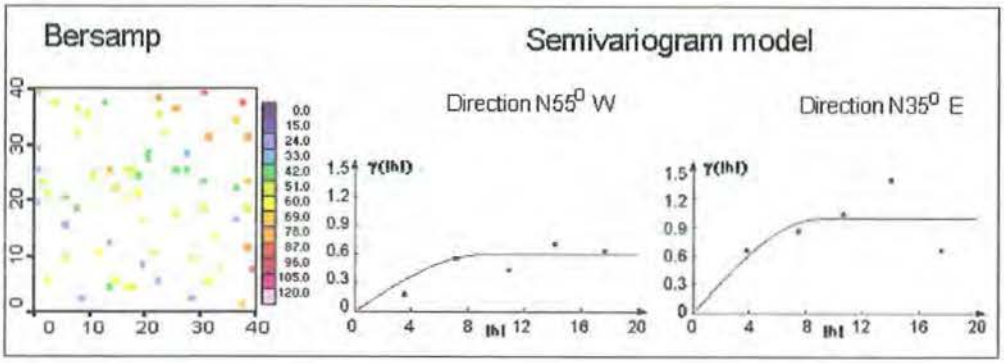


Figure 21: Post plot of *Bersamp* and semivariogram models fitted to experimental semivariograms for normal scores.

The normality of the samples discussed in this Chapter is necessary but not sufficient to ensure that the underlying random function follows a multivariate normal distribution. Therefore, before applying the algorithms to the above samples, the multiGaussian assumption is checked for the standard normal scores. Since it is not possible to fully check the normality of more than two-point cumulative distribution functions, in practice one can only check the biGaussian assumption. If the biGaussian assumption is valid, then the multiGaussian assumption is taken to be satisfied. The checking of the biGaussian assumption is shown in Appendix 12.2.

In the isotropic case the results do not indicate any violation of the biGaussian

assumption. In the anisotropic case, the hypothesis on the biGaussian is not strictly satisfied since the semivariograms in Figures 67 and 68 indicate that the high valued data are more correlated than the low values. However, since the violation is not too severe, we do not reject the biGaussian hypothesis and assume that the anisotropic samples come from a multivariate standard normal distribution.

6 SINGLE-LEVEL WAVELET-BASED CONDITIONAL SIMULATION

In this Chapter we describe our first two conditional simulation algorithms that make use of wavelets. They are both single-level conditional simulation algorithms and are named HSIM (using the Haar basis) and DB2SIM (using the Db2 basis). In both algorithms, the conditioning data and the values to be simulated in a study region are regarded as the scaling coefficients at the original scale 2^0. HSIM and DB2SIM then make use of the location independence of wavelet and scaling spectra to simulate the wavelet and (or) scaling coefficients at the finest scale 2^1. The simulated values are computed from the simulated scaling and wavelet coefficients using the inverse discrete wavelet transform. Both algorithms are based on the assumptions that the underlying random function is second order stationary and that the sample is from a standard normal distribution. Therefore, if the assumption of normality is not satisfied, the application of each algorithm needs to be preceded by the transformation of the conditioning data into standard normal scores before the modelling of the semivariogram. The covariance model is then computed from the semivariogram model using $C(\mathbf{h}) = 1 - g(\mathbf{h})$. The results in normal score space are then transformed back into the attribute values after the conclusion of the algorithms.

6.1 The Algorithm HSIM

The algorithm HSIM is a sequential single level wavelet-based conditional simulation algorithm based on Haar wavelets. The simulation at each location in the study region is carried out in a window whose size is determined by the filter length with the movement on a random path. Since the Haar filters contain only two non-zero

coefficients, in the two-dimensional case the size of the window is 2×2. At each location of the window the wavelet coefficients at the finest scale 2^1 are simulated and the associated scaling coefficient is computed in terms of the simulated wavelet coefficients and the conditioning value(s) within the window. The simulated values in the window are obtained via the inverse discrete wavelet transform. The reason for starting the simulation at the finest scale is to associate the inverse discrete wavelet transform, the wavelet and scaling coefficients at the finer scale 2^1 with the data at the original scale 2^0.

6.1.1 Description of the Algorithm HSIM

The algorithm is carried out as follows. Given a window of size 2×2 the method proceeds by first determining the number of sample values and their locations within the window and then simulating the wavelet coefficients. If there are l values, where $l = 1, 2$ or 3, within the window then we simulate $4-l$ wavelet coefficients, compute the associated scaling coefficient and then calculate the values at the remaining locations in the window. The algorithm is outlined as follows:

Step 1: Given a sample that is normally distributed with mean 0 and variance 1 and with covariance function $C(\mathbf{h})$, we compute each Haar wavelet spectrum according to equations (114)-(116) for $j = 1$.

Step 2: Generate a random path visiting all grid nodes in the study region exactly once.

Step 3: Move to the first grid node.

Step 4: Construct a window of size 2×2 containing the node. If the node does not lie on the right or on the bottom boundary of the study region, it is located at

row 1 and column 1 of the window. If the node lies on the right boundary, it is located at row 1 and column 2 of the window. If the node lies on the bottom boundary, the node is located at row 2 column 1, and if the node lies on the right and bottom boundary, the node is located at row 2 and column 2 of the window. Denote the values, which are regarded as the scaling coefficients at scale 2^0, at these locations by $a[u,v]$, with $u,v = 0,1$. If the window does not contain any sample locations or previously simulated nodes, or if all nodes in the window have already been assigned values, move to the next node. If the window contains l, where $l = 1,2$ or 3, known values then denote the three wavelet coefficients and the scaling coefficient at scale 2^1 associated with the window by d_1^1, d_2^1, d_3^1 and c^1 respectively. Hence, each value in the window is expressed as a linear combination of the wavelet and scaling coefficients via the inverse discrete wavelet transform using equation (92). Here, in the case of Haar wavelets, the system of linear equations is expressed as

$$\begin{bmatrix} .5 & .5 & .5 & .5 \\ .5 & .5 & -.5 & -.5 \\ .5 & -.5 & .5 & -.5 \\ .5 & -.5 & -.5 & .5 \end{bmatrix} \times \begin{bmatrix} c^1 \\ d_1^1 \\ d_2^1 \\ d_3^1 \end{bmatrix} = \begin{bmatrix} a[0,0] \\ a[0,1] \\ a[1,0] \\ a[1,1] \end{bmatrix}. \tag{127}$$

Since the number of sample and previously simulated locations is l, where $l = 1,2$ or 3 then, in (127), l values of $a[.,.]$ are known. The remaining $4-l$ unknown values of $a[.,.]$ are those that need to be simulated. By putting the known values $a[.,.]$ into their associated rows on the right hand side of (127), two systems of linear equations are extracted. The first system consists of l equations that express the known values in the window in terms of one scaling and three wavelet coefficients c^1 and $d_{k,k=1,2,3}^1$. This system is used to simulate

the scaling and wavelet coefficients c^1 and $d^1_{k,k=1,2,3}$. The second system consists of $4-l$ linear equations that express the unknown values to be simulated in terms of c^1 and $d^1_{k,k=1,2,3}$. This system is used to compute the simulated values in the window after c^1 and $d^1_{k,k=1,2,3}$ are obtained using the first system. For example, if $l=2$ and $a[0,0]$ and $a[1,1]$ are known then, according to locations of these nodes in the window, from system (127) the first and the second systems are

$$\begin{aligned} .5c^1 + .5d^1_1 + .5d^1_2 + .5d^1_3 &= a[0,0] \\ .5c^1 - .5d^1_1 - .5d^1_2 + .5d^1_3 &= a[1,1] \end{aligned} \tag{128}$$

and

$$\begin{aligned} .5c^1 + .5d^1_1 - .5d^1_2 - .5d^1_3 &= a[0,1] \\ .5c^1 - .5d^1_1 + .5d^1_2 - .5d^1_3 &= a[1,0]. \end{aligned} \tag{129}$$

Step 5: Use the first system to obtain the wavelet coefficients $d^1_{k,k=1,2,3}$ and the scaling coefficient c^1. Since this system consists of l linear equations in 4 unknowns, we simulate $4-l$ wavelet coefficients $d^1_k, k=1,2,3$ from $N(0,\zeta^1_k)$ as defined in (111) to reduce the first system to a system of l equations in l unknowns. The reduction is achieved by substituting the simulated wavelet coefficients into their associated locations in the first system and rearranging these equations. The scaling coefficient c^1 and the remaining $(l-1)$ wavelet coefficients are obtained by solving the system. If $l=1$, then all three wavelet coefficients are simulated and the scaling coefficient c^1 is computed in terms of the simulated wavelet coefficients and the conditioning value. If $l=3$, only one wavelet coefficient is simulated. The scaling coefficient and the other two wavelet coefficients are

computed in terms of the simulated wavelet coefficient and the conditioning values. Since there are three types of wavelet coefficients but only one is drawn, the wavelet coefficient which needs to be simulated is chosen at random. If $l = 2$ as in the case of the given example, because the system (128) has 2 linear equations in 4 unknowns, two wavelet coefficients are simulated. These two wavelet coefficients are chosen so that the coefficient matrix of the reduced system is not singular. For this particular example d_1^1 and d_3^1 are drawn and by substituting the values of the two simulated coefficients into (128) and rearranging the equations, the system (128) becomes

$$\begin{aligned} .5c^1 + .5d_2^1 &= a[0,0] - .5d_1^1 - .5d_3^1 \\ .5c^1 - .5d_2^1 &= a[1,1] + .5d_1^1 - .5d_3^1. \end{aligned} \tag{130}$$

The coefficients c^1 and d_2^1 then are computed by solving the reduced system (130).

Step 6: Substitute the values of c^1 and $d^1_{k,k=1,2,3}$ into the second system to compute the $4 - l$ simulated values at unknown locations. In the above example, this step is carried out by substituting the values c^1, d_1^1, d_2^1 and d_3^1 into system (129) to compute the simulated values $a[0,1]$ and $a[1,0]$.

Step 7: Add the newly simulated values to the conditioning values and the window moves to next location in the random path

Step 8: Repeat steps 4-7 until the window reaches the end of the path.

Step 9: Go back to step 3 until all locations have been simulated.

6.1.2 Applications of HSIM

The performance of HSIM is tested for isotropic and anisotropic data by using the samples *pH100IRn* and *B64IRn* respectively. The result of the simulation is visually assessed by comparing the mosaic plots of the resulting realisations with the mosaic plots of the exhaustive data sets *pHsoiln* and *Berean*. In each case the mosaic plots of the realisations give some sharp-edged patches as shown in the middle column of Figure 22. This artefact is more pronounced in the anisotropic case. Spatial continuity is assessed by comparing the experimental semivariograms of the realisations with the associated semivariogram models. For the isotropic sample, the experimental semivariogram for the realisation is similar to the semivariogram model, however, with a shorter range. For the anisotropic sample, the semivariogram model is not adequately reproduced.

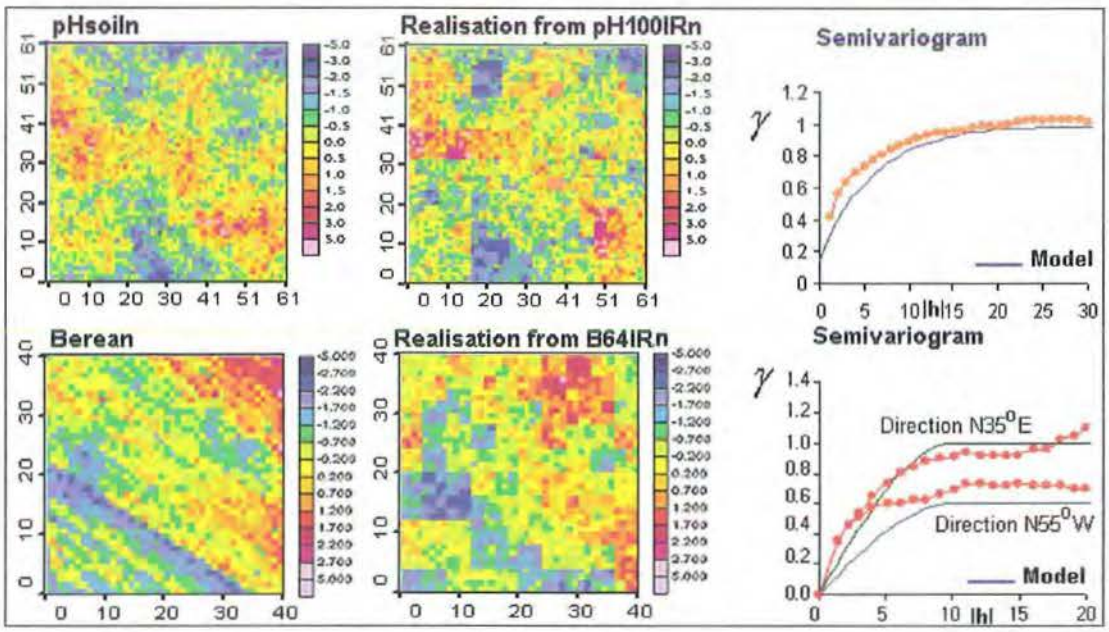


Figure 22: Reference images (left), realisations and associated experimental semivariograms (middle and right).

Our conjecture is that the artefact found in the mosaic maps and the poor reproduction of the spatial continuity in HSIM are caused by the nature of the Haar filters. Since the Haar filters contain only two non-zero coefficients, the size of the window may simply be too small to capture the spatial continuity. Excluding the previously simulated values, the window usually contains at most one conditioning value. In addition, in HSIM, the covariance model is only used to compute the variances of wavelet coefficients (the wavelet spectra). Any correlation between the locations of the simulated values and data locations in the neighbourhood is ignored. The result is that the simulated values in a window usually depend only on one conditioning value in this window. If this value is large, the simulated values are large and conversely small values typically lead to small simulated values. Therefore, discontinuity and patchiness are produced when the window moves from a high value location to a low value location.

As the window of size 2×2 does not contain enough data to fully capture the spatial continuity, we have attempted to expand the window size to a multiple of the Haar filter length, for example 4×4, 6×6, 8×8 and so on, to include more nodes. However, because of the discontinuity of Haar wavelets, this attempt increases the computational effort but the outcome of the simulation is similar to that in the case the window of size 2×2. Therefore, we deduce that the result of single level wavelet-based conditional simulation can only be improved if a different wavelet basis is used or the wavelet method is used in combination with other methods.

6.2 The Algorithm DB2SIM

In order to overcome the problem with HSIM detailed in Section 6.1, we moved to develop the algorithm called DB2SIM (Tran et al, 2001). In DB2SIM, the Haar basis

is replaced by the Db2 basis and the location independence of the scaling spectrum and the wavelet spectrum is used in combination with kriging to capture the spatial continuity. The simulation is again carried out sequentially in a window moving along a random path.

Although the Db2 filters contain four non-zero coefficients, the size of the window in this algorithm is not determined based on the filter length. We have attempted to use a window of size 4×4, however, because of the nature of the Db2 filters this attempt has led to problems through ill-conditioning of the resulting matrices. In addition, if the wavelet method is used alone, the spatial continuity is still not fully captured since the size of the window is still small compared with the study region.

Therefore, in DB2SIM the size of the window used is 2×2 as in the case of the Haar wavelets. In each window the associated wavelet and scaling coefficients at the finest scale 2^1 are simulated and the simulated values are computed via the inverse discrete wavelet transform. To capture the spatial continuity, the ordinary kriging estimates using the sample (assume that the data follow a standard normal distribution) are used as the training image for the simulation.

6.2.1 Description of the Algorithm DB2SIM

The conditional simulation algorithm DB2SIM comprises the following steps.

Step 1: Given a sample from a standard normal random function (if this condition is not met then the sample needs to be transformed into normal score space), obtain the covariance function $C(\mathbf{h})$ from the semivariogram model via the expression $g(\mathbf{h}) = C(0) - C(\mathbf{h})$ and compute the scaling and wavelet spectra at the finest scale 2^1 using equations (113)-(116) for $j = 1$.

Step 2: Compute the kriging estimates at all unestimated nodes from the sample

using the same semivariogram model. That is, if the conditioning data follow a standard normal distribution then the kriging estimates are obtained using the data and the semivariogram model fitted to the experimental semivariogram for the data. If the conditioning data do not follow a standard normal distribution then the kriging estimates are obtained using the normal scores and the semivariogram model fitted to the experimental semivariogram for the normal scores.

Step 3: Generate a random path visiting each grid node in the study region exactly once.

Step 4: Move to the first grid node on the path.

Step 5: Construct a window of size 2×2 containing the node. The location of the node in the window is determined in the same way as for HSIM. Denote the values at these nodes by $a[u, v]$, with $u, v = 0, 1$. If all locations in the window are known, move to the next location on the path. If the window contains unestimated values then the location to be simulated is the first unestimated location in the window according to row-ascending order. The kriging estimates are assigned to the other l, where $l = 0, 1, 2$ or 3 unknown locations in the window so that the node whose value to be simulated is the only unknown location in the window. Each value $a[u, v]$ in the window can be expressed as a linear combination of scaling and wavelet coefficients via the inverse discrete wavelet transform in equation (92). Here, for Db2 wavelets, because of the edge effect each value in the window is expressed in terms of 16 wavelet and scaling

coefficients by

$$
\begin{aligned}
a[u,v] &= \sum_{m=-1}^{0}\sum_{n=-1}^{0} h[u-2m]h[v-2n]c[m,n] \\
&+ \sum_{m=-1}^{0}\sum_{n=-1}^{0} g[u-2m]h[v-2n]d_1[m,n] \\
&+ \sum_{m=-1}^{0}\sum_{n=-1}^{0} h[u-2m]g[v-2n]d_2[m,n] \\
&+ \sum_{k=-1}^{0}\sum_{n=-1}^{0} g[u-2k]g[v-2n]d_3[k,n]. \qquad (131)
\end{aligned}
$$

Since three values in the window are known, by putting these values into (131), two systems of linear equations are extracted. The first system consists of three linear equations that express the known values in terms of 16 scaling and wavelet coefficients $c[m,n]$ and $d_k[m,n]$, where $m,n = -1,0$ and $k = 1,2,3$. These three equations are used to simulate and then to compute the scaling and wavelet coefficients $c[.,.]$ and $d_k[.,.]$. The second system consists of one linear equation that expresses the unknown value in terms of $c[.,.]$ and $d_k[.,.]$. This equation is used to compute the unknown value after all 16 coefficients $c[.,.]$ and $d_k[.,.]$ are obtained.

Step 6: Simulate 13 scaling and wavelet coefficients of $c[.,.]$ and $d_k[.,.], k = 1,2,3$ in the first system from the distributions described in (111) and (112). Since there are three equations in 16 unknowns, 13 coefficients are simulated and the other three are calculated in terms of the known values and simulated coefficients. The three coefficients, which need to be calculated, are selected so that the matrix of the filter coefficients corresponding to these coefficients is not ill-conditioning or singular. For example, if the three coefficients to be calculated are $c[-1,-1], d_1[0,0]$ and $d_2[0,-1]$, and $K_i, i = 1,2,3$ are the sums computed

by substituting the values of the 13 simulated coefficients into their corresponding locations in each equation of system (131), then by rearranging the three equations we obtain a system of three linear equations in three unknowns

$$h[u+2]h[v+2]c[-1,-1] + g[u]h[v]d_1[0,0] + h[u]g[v+2]d_2[0,-1] = a[u,v] - K_i. \tag{132}$$

Hence, the three coefficients $c[-1,-1], d_1[0,0]$ and $d_2[0,-1]$ are obtained by solving system (132). As the kriging estimates implicitly carry the spatial correlation of the simulated process, the simulated wavelet and scaling coefficients, which honour the kriging estimates, also implicitly capture this spatial correlation. Having obtained all 16 wavelet and scaling coefficients, the unknown value in the window is computed by substituting these coefficients into the second system. The kriging estimates are discarded before the next location is simulated.

Step 7: Move to the next location in the random path and repeat steps 5 and 6 until all locations have been simulated.

6.2.2 Applications of DB2SIM

Four samples are used to assess the performance of DB2SIM, *pH100n* and *pH100IRn* for the isotropic case and *B64n* and *B64IRn* for the anisotropic case. The mosaic plots of the realisations from the four samples together with the associated semivariograms and the quantile-quantile plots of the realisations against the associated samples are shown in Figure 23. In each case, DB2SIM approximately reproduces the histograms and the semivariogram models of the associated samples, regardless of the sampling method, even though the minimum of each realisation is slightly lower than

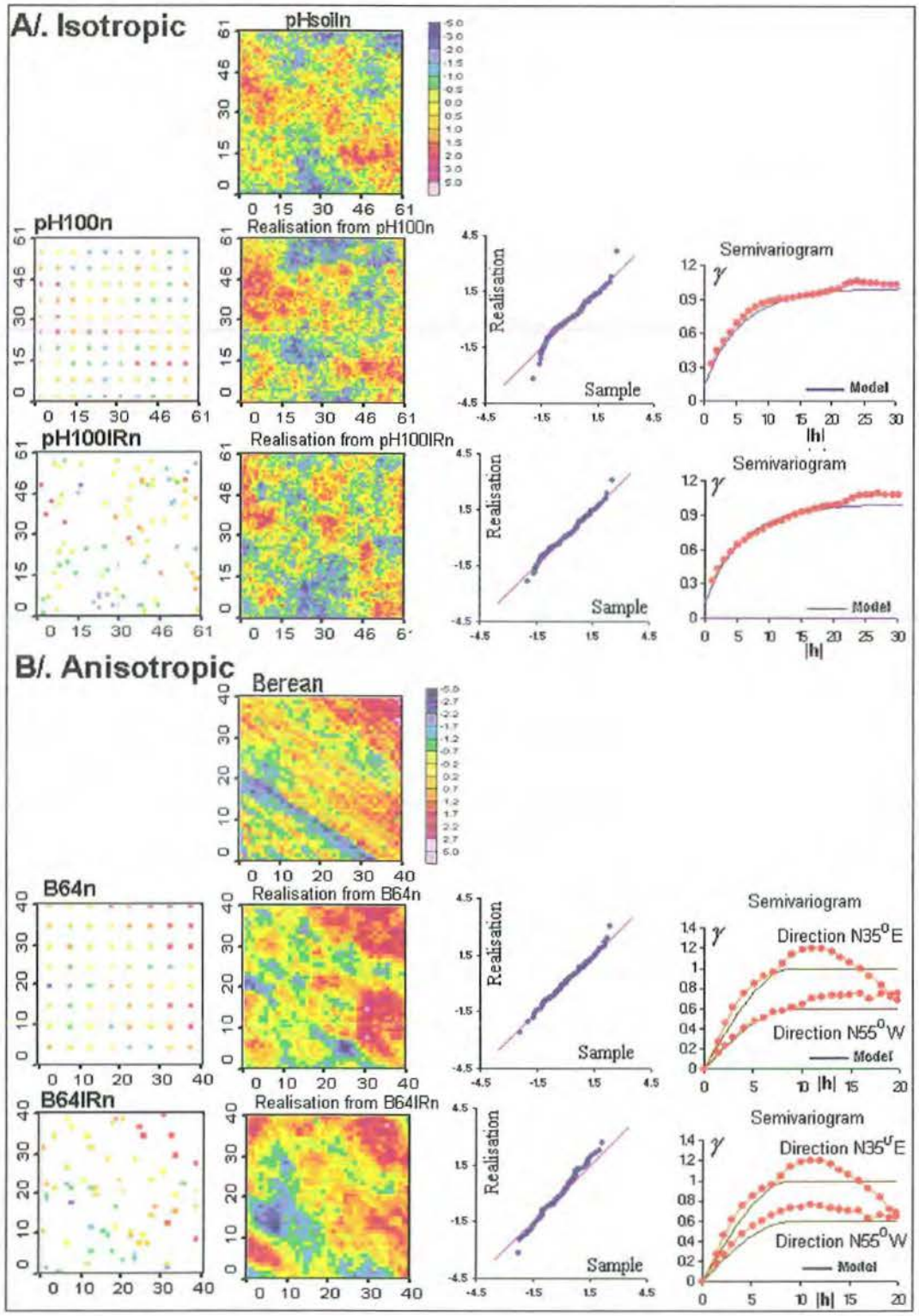


Figure 23: Mosaic plots of typical realisations simulated via DB2SIM, Q-Q plots, experimental semivariograms and mosaic plots of associated exhaustive data sets and post plots of corresponding samples.

the corresponding sample minimum and the maximum is higher than the corresponding sample maximum. It can be seen from the mosaic plots that the realisations have captured the features of the corresponding exhaustive data set *pHsoiln* and *Berean* and the associated samples. No artefacts are found in the mosaic plots.

Even though DB2SIM reproduces the sample histogram and the semivariogram model with any spatial structure, it has to make use of the kriging estimates to do so. Therefore, the result of the algorithm will depend on the parameters used to obtain the kriging estimates. In addition with the estimation of the training image, although carried out only once, DB2SIM requires the solving of one system of three linear equations with 16 unknowns for each simulated value. Because of this, the computational efficiency of using the wavelet method is diminished.

6.3 Comments on HSIM and DB2SIM

Based on the performance assessment in this Chapter, both HSIM and DB2SIM possess some drawbacks. In HSIM, the simulated realisations approximately reproduce the associated sample histogram and semivariogram model in the isotropic case but fail to adequately reproduce the semivariogram model in the anisotropic case. In addition, artefacts may be found in the mosaic plots of the realisations. For DB2SIM, even though the realisations reproduce the sample histogram and semivariogram model regardless of spatial structure and sampling method, this achievement comes with some loss of the speed advantage of using the wavelet method. In addition, over-conditioning by using the kriging estimates twice reduces the differences among realisations.

Despite the shortcomings of HSIM and DB2SIM, the results from these algorithms indicate that one can use the wavelet method to obtain reasonable conditional simu-

lations. For both algorithms the drawbacks are caused by the fact that the window used to capture the conditioning data is small compared with the study region. If the window can contain more information from the study region, the problems can be overcome. However, due to the localisation property of wavelets, increasing the window size is not a feasible solution.

Therefore, instead of increasing the window size so that it is large enough compared with the study region one can compress the study region so that it is small enough compared with the window. Then the window can capture spatial continuity of the study region. In order to do so, the discrete wavelet transform must be applied several times. Hence, the simulation must start at some scale coarser than scale 2^1 and the process of reconstructing the original scaling image is not carried out in just one level but via many levels. This approach is used in the multi-level wavelet-based conditional simulation algorithms in Chapter 7.

7 MULTI-LEVEL CONDITIONAL SIMULATION ALGORITHMS

The single level simulation algorithms using the wavelet bases discussed in Chapter 6 start the simulation at scale 2^1, one level coarser than the original image. These algorithms did not exploit the fact that the scaling image at the coarser scale, obtained via the discrete wavelet transform, is a compression of the scaling image at the previous finer scale. On compressing the study region, the discrete wavelet transform also compresses the spatial continuity. Hence, if the discrete wavelet transform is applied several times, the study region and the corresponding spatial continuity in this region will be compressed to a level from which the size of the window, determined by the corresponding wavelet filter length, can capture the spatial continuity.

Therefore, in order to improve the results and to save computational time, we explored the possibility of starting the simulation at some scale coarser than 2^1. Our two approaches for an effective and efficient multi-level simulation algorithm are PWSIM and DWSIM both of which are detailed in this Chapter. All of the assumptions and preparation steps made in the case of single-level simulation algorithms remain applicable for these multi-level algorithms. That is, PWSIM and DWSIM are based on the assumptions that the data are from a multivariate standard normal distribution and that the underlying random function is second-order stationary. If the assumption of normality is not satisfied, the normal score transformation is applied to the conditioning data before modelling the semivariogram.

7.1 The Algorithm PWSIM

Our multi-level wavelet-based conditional simulation algorithm PWSIM is obtained by post-conditioning a multi-level nonconditional simulation algorithm. The nonconditional simulation method exploited here is that introduced by Zeldin and Spanos (1995). This method has been applied in engineering context to simulate Gaussian stationary random functions. Zeldin and Spanos used it to simulate realisations of a one-dimensional random function whose autocorrelation function is

$$\rho(x_1, x_2) = \cos(\omega |x_1 - x_2|) e^{-\frac{|x_1 - x_2|}{\alpha}}$$

or of a two-dimensional random function whose autocorrelation function is of the form

$$\rho(x_1, x_2, y_1, y_2) = e^{-\frac{\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}}{\alpha}}$$

where α and ω are scalars. We shall denote this algorithm by WSIM. For two-dimensional random functions, the wavelet basis used in WSIM can be either Haar wavelets or Daubechies wavelets with up to three vanishing moments. This nonconditional simulation algorithm starts the simulation at some coarse scale then recursively builds up the simulated realisations (at the original scale) via the inverse discrete wavelet transform.

7.1.1 Description of the Algorithm PWSIM

The algorithm WSIM makes use of the fact that after each application of the discrete wavelet transform the size of a set of evenly-spaced discrete values is reduced, in one dimension by a factor of two and in two dimensions by a factor of four. Therefore, after the discrete wavelet transform has been applied a number of times, at some coarse scale the size of the scaling coefficient image is small. Consequently, the size

of the covariance matrix for the associated scaling coefficients is also small enough to be decomposed into a product of an upper and a lower triangular matrices.

As the initial step, the covariance matrix K of the associated scaling coefficients at the scale 2^j is decomposed into lower and upper triangular matrices using the Cholesky decomposition (Gerald and Wheatley, 1994), that is

$$K = LU$$

where

$$L^T = U.$$

Then the associated scaling coefficients are simulated by setting

$$c^j = Lw \tag{133}$$

where w is a vector of random numbers drawn from a standard normal distribution.

The wavelet coefficients at this scale are simulated using the correlation between wavelet and scaling coefficients at the same scale. In order to do so, each wavelet coefficient at scale 2^j is expressed as a linear combination of scaling coefficients at the same scale plus a noise. In two-dimensional space, each wavelet coefficient of a stationary random function is simulated in terms of scaling coefficients at the same scale by

$$d_k^j[m,n] = \sum_{k_2,l_2=-i}^{i} \alpha_{k(k_2,l_2)}^j c^j[m+k_2, n+l_2] + \gamma_k^j R_k(m,n) \tag{134}$$

for $k = 1,2,3, i = 1,2...$ where for each k the noise $R_k(m,n)$ is randomly drawn from a standard normal distribution and is statistically independent of $c^j[.,.]$, and since the covariance of two scaling and or wavelet coefficients at the same scale decreases in an inverse proportion with the difference of the indices of these coefficients (Vannucci

and Corradi, 1997), we choose $i = M$, the number of vanishing moments to save computational effort. Based on (134), each wavelet coefficient $d_k^j[m,n]$ is a linear combination of $(2i+1)^2$ scaling coefficients $c^j[.,.]$ centered at $c^j[m,n]$ plus a random noise.

It can be shown, by multiplying both sides of equation (134) by $c^j[m+k_1, n+l_1]$, where $k_1, l_1 = -i, ..., 0, ..., i$ and taking the expected value of each side, that the weights $\alpha^j_{k(k_2,l_2)}$ are the solutions of the system

$$\sum_{k_2,l_2=-i}^{i} \alpha^j_{k(k_2,l_2)} \beta^j_{(m+k_2,m+k_1),(n+l_2,n+l_1)} = \eta^j_{k(m,m+k_1),(n,n+l_1)}. \tag{135}$$

By squaring both sides of equation (134) and taking the expected value of each side, the value γ_k^j is computed by

$$\gamma_k^j = \sqrt{\tau^j_{k(m,n),(m,n)} - \sum_{k_2,l_2=-i}^{i} \sum_{k_1,l_1=-i}^{i} \alpha^j_{k(k_2,l_2)} \alpha^j_{k(k_1,l_1)} \beta^j_{(m+k_2,m+k_1),(n+l_2,n+l_1)}}. \tag{136}$$

Having simulated the wavelet coefficients associated with the scaling image at scale 2^j, the scaling image at the finer scale 2^{j-1} is reconstructed via the inverse discrete wavelet transform. The process is carried out recursively by computing the associated wavelet coefficients at scale 2^{j-1} and applying the inverse discrete wavelet transform to obtain the scaling coefficients at scale 2^{j-2} until the original scale is reached.

Note that, since WSIM is based on the inverse discrete wavelet transform, the size of the simulated realisation must be a power of 2.

The algorithm WSIM is summarised in Figure 24 and an illustration for the simulation of the wavelet coefficient $d_1^j[m,n]$ for $m = 2, n = 3$ and $i = 2$ with the scaling image of size 8×8 is given in Figure 25. The location of the wavelet coefficient $d_1^j[2,3]$ is indicated by the shaded cell in Figure 25A. This coefficient will be simulated by a linear combination of 25 scaling coefficients (at the same scale) in the shaded cells in

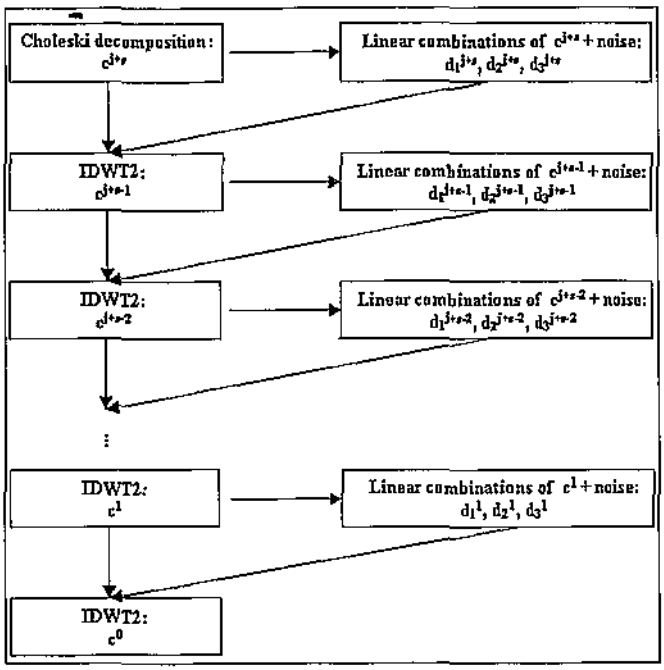


Figure 24: Summary of WSIM.

Figure 25B plus a noise. The weights of the linear combination are the solutions of system (135) and the standard deviation of the noise is computed via equation (136).

Since the covariance of two scaling or wavelet coefficients is independent of the coefficient indices, each move from a coarse scale to the previous finer scale involves the solving of the system (135) and equation (136) once for every k, that is, three systems of $(2i+1)^2$ linear equations plus three equations.

The computation of the weights must be preceded by the computation of the covariance tables at the coarser scales. If the size of the study region to be simulated

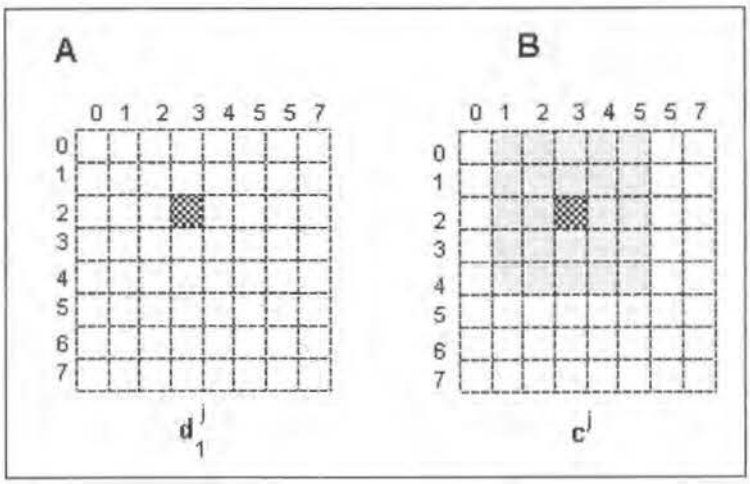


Figure 25: A/. Wavelet coefficient to be simulated (shaded); B/. Scaling coefficients used to simulate wavelet coefficient in A (shaded).

is $P \times Q$, the computation of the covariance tables for the coarser scales is initialised by computing the covariance table at the original scale of size $(2P+1) \times (2Q+1)$. Then the covariance table at scale $2^j, j > 0$ is computed from the covariance at the previous finer scale 2^{j-1}. Upon moving from one scale to the next coarser scale, the size of the covariance table reduces. In general, at scale $2^j, j > 0$ the size of the covariance table is

$$(2\frac{P}{2^j}+1) \times (2\frac{Q}{2^j}+1).$$

The number of algebraic operations used to compute the covariance values for the coarser scales depends on the wavelet basis. In two dimensional space, according to formula (99), for wavelets with M vanishing moments, in order to compute each value in a covariance table at the coarser scale one needs to carry out $4 \times (2M)^4$ multiplications and $(2M)^4 - 1$ additions. Therefore, the higher the number of vanishing moments, the more computation time is required. Detail on the computational effort will be given in Appendix 12.3.

We then condition the non-conditional simulated realisation obtained by WSIM by adding to each non-conditional simulated value $z_s(\mathbf{u})$ the error between the ordinary kriging estimate $z_K^*(\mathbf{u})$ using the actual data and the ordinary kriging estimate $z_{sK}^*(\mathbf{u})$ using the simulated values at data locations, that is

$$z_{sc}(\mathbf{u}) = z_s(\mathbf{u}) + [z_K^*(\mathbf{u}) - z_{sK}^*(\mathbf{u})] \tag{137}$$

Since the configuration between data locations as well as the covariance model is the same for the two kriging images, this step involves solving one kriging system for each simulated value. PWSIM is our final result, combining WSIM with the post conditioning described above.

7.1.2 Case Study 1: Applications of PWSIM to the Isotropic Case

The performance of PWSIM in the isotropic case was tested by applying PWSIM to samples *pH100n* and *pH100IRn*, using each of the Haar and the Db2 bases. Since PWSIM requires that the size of the simulated map be a power of 2, and since the size of the study region in this case is 61×61, the size of the region to be simulated was extended to a size of 64×64. By starting the simulation at scale 2^4 with the initial scaling image of size 4×4, for each wavelet basis and each sample 1000 nonconditional realisations were simulated. The dimensions of each simulated realisation were then trimmed to 61×61 before the application of ordinary kriging to condition the data. Based on the range of our semivariogram models and the size of the study region, a search radius of 20 for the data with maximum of 16 data is used in the estimation.

The performance of the simulation, in general, is visualised by comparing the mosaic plots of three realisations, randomly selected from the sets of 1000 realisations from *pH100n* and *pH100IRn* simulated via each of the wavelet bases, with the mosaic plot of the data set *pHsoiln* and the post plots of the corresponding samples. The

spatial continuity reproduction is visualised by comparing the experimental semivariograms of these realisations with the semivariogram models. The sample histogram reproduction is visualised via the quantile-quantile plots of the realisations against the corresponding samples. The mosaic plots for these realisations, together with the mosaic plot of *pHsoiln* and the post plots of *pH100n* and *pH100IRn*, are shown in Figure 26. Realisations from *pH100n* simulated using the Haar basis and the Db2 basis are shown in columns 1 and 2, respectively, while those from *pH100IRn* are shown in columns 3 and 4.

The mosaic plots for the simulated realisations in Figure 26 indicate that, regardless of the wavelet basis used, the main features of the data set *pHsoiln* and of the associated samples have been captured. At locations where *pHsoiln* and the samples have low values the simulated realisations also have low values. At locations where *pHsoiln* and the samples have high values the simulated realisations also have high values. No artefacts are seen in any of the mosaic plots for the realisations.

Experimental semivariograms of these realisations in Figure 27 show that for both wavelet bases the semivariogram models of both samples are reproduced. The experimental semivariograms of the realisations from sample *pH100n* in columns 1 and 2 and those for realisations from *pH100IRn* in columns 3 and 4 of Figure 27 have captured the behaviour of the associated semivariogram model. In particular, the semivariogram of Realisation #2 from *pH100n*, simulated using the Db2 basis, in column 2 fits the model very well.

The summary statistics and the quantile-quantile plots in Figure 28 show that the histograms of the two samples are approximately reproduced. The means and variances of the realisations are close to those of *pH100n* and *pH100IRn*. For all realisations the extreme values fall outside the range of the corresponding samples. This

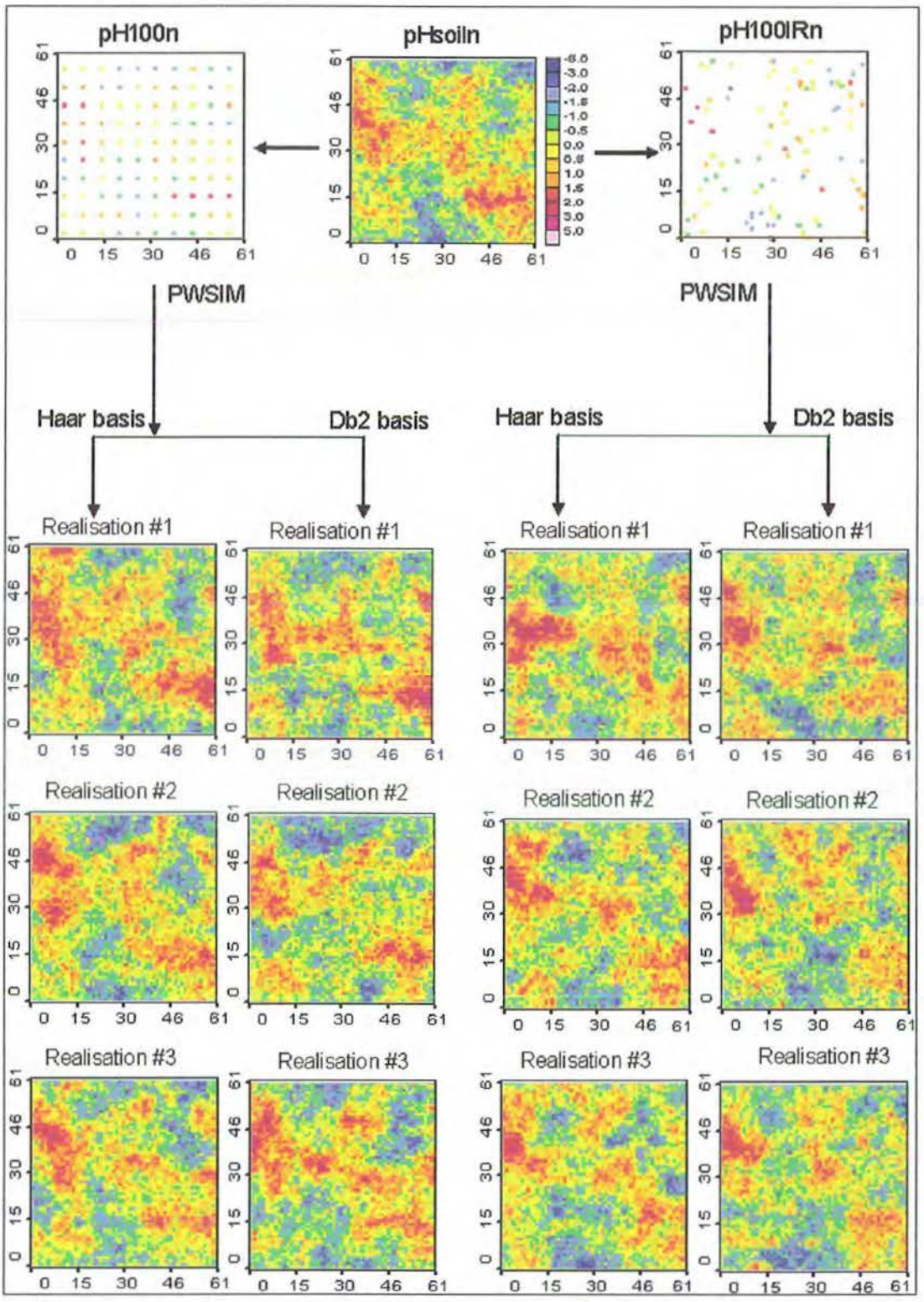


Figure 26: Mosaic plot of *pHsoiln*, post plots of *pH100n* and *pH100IRn* compared with three randomly selected realisations simulated via PWSIM.

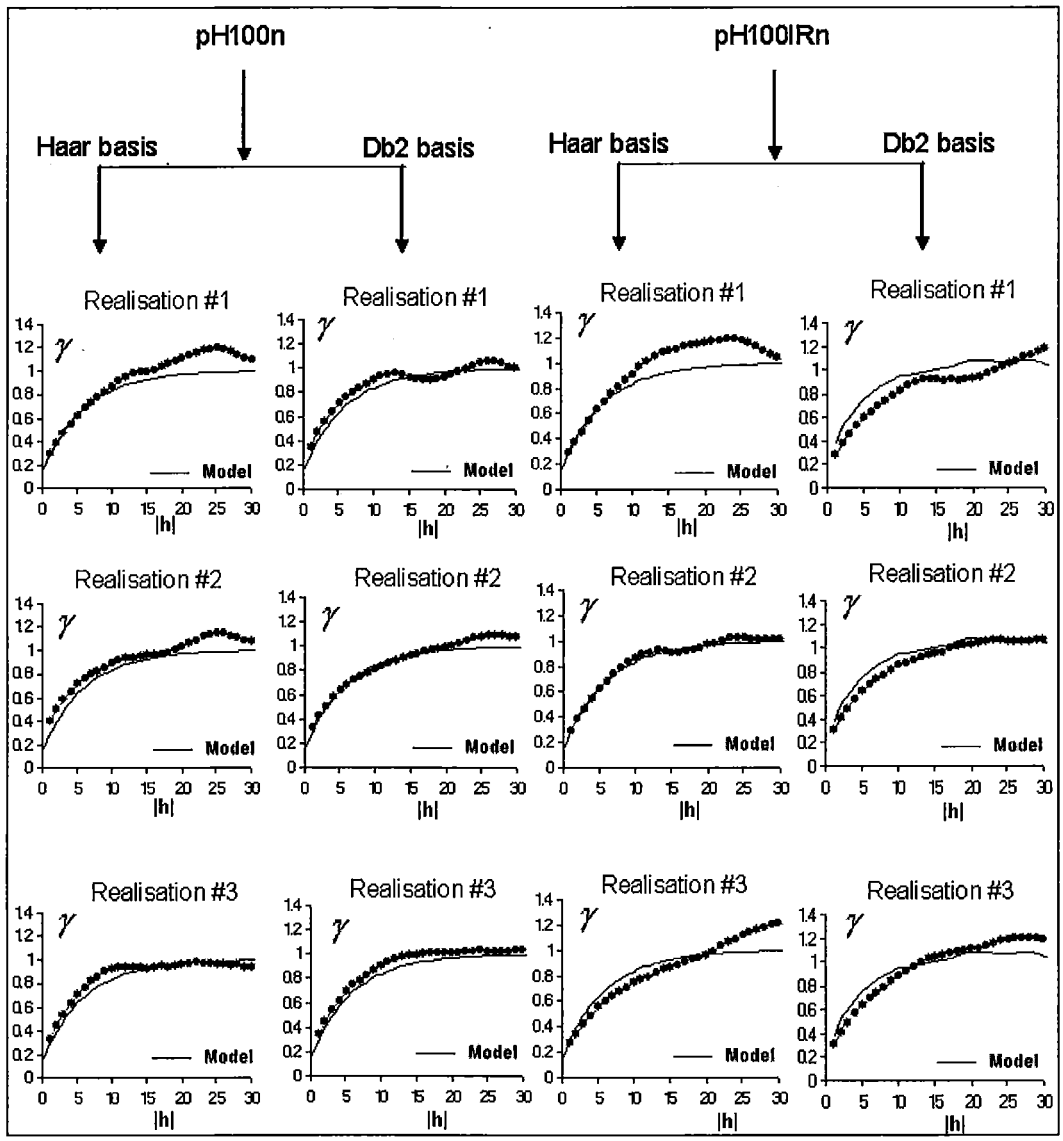


Figure 27: Experimental semivariograms of the three randomly selected realisations from *pH100n* and *pH100IRn* simulated via PWSIM.

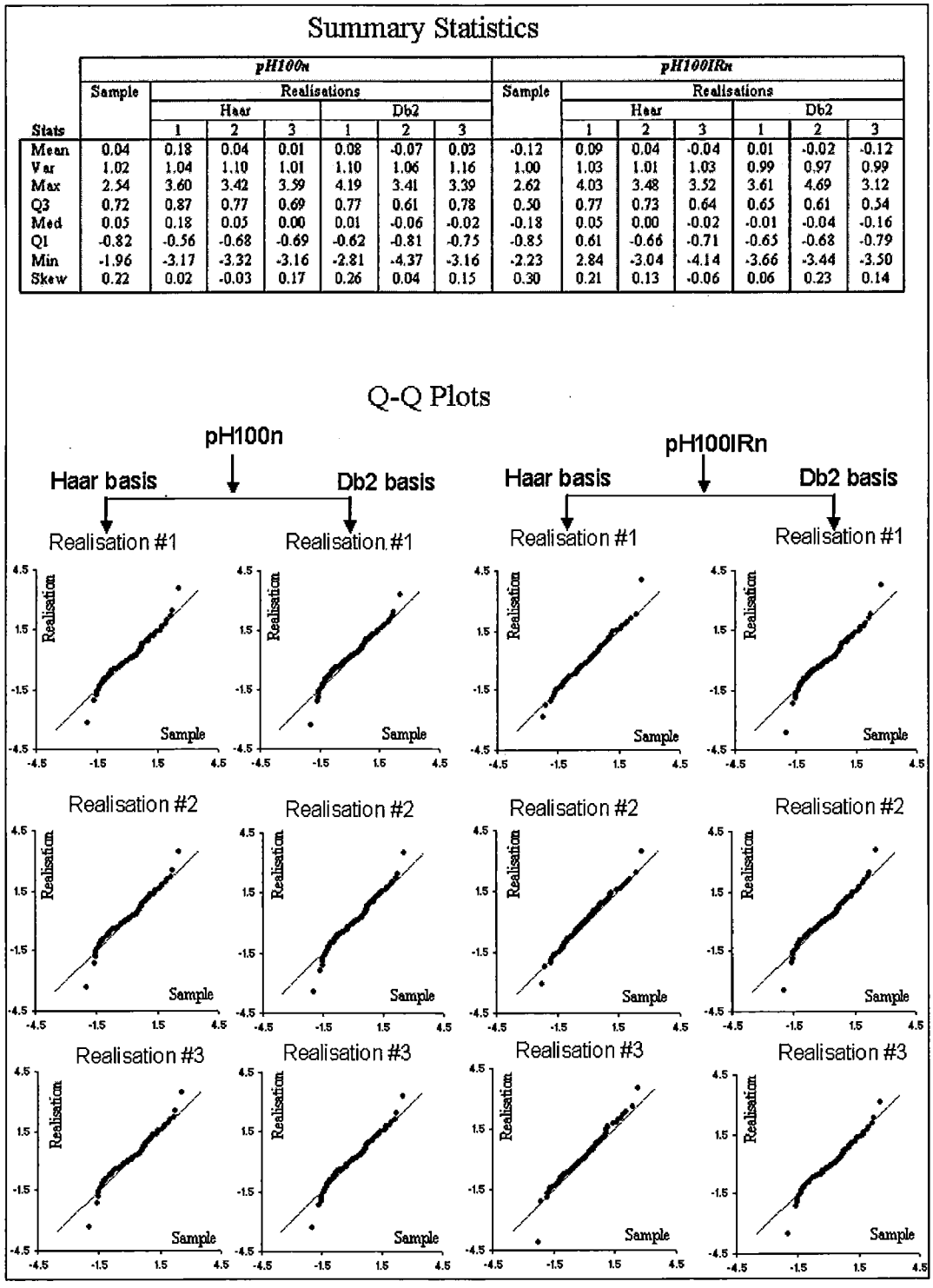


Figure 28: Summary statistics and Q-Q plots of three randomly selected realisations from pH100n and pH100IRn (simulated via PWSIM) against corresponding samples.

feature of the simulation algorithm merely reflects the fact that Monte Carlo simulation is used and no range restrictions are made. In fact, in most cases a restriction to the range implied by the sample is avoided and to ensure that the extreme values of the realisations fall outside the range. Except for those of realisations from *pH100n* simulated using the Db2 basis, which fluctuate about the sample skewness coefficient, the skewness coefficients of the other realisations are lower than the skewness coefficient of the associated sample. Compared with those in the case of the Haar basis, the realisations obtained via the Db2 basis have more low values.

7.1.3 Case Study 2: Applications of PWSIM to the Anisotropic Case

We now investigate the performance of PWSIM in the case of the anisotropic samples *B64n* and *B64IRn*. By starting the simulation at scale 2^4 with the initial scaling images of size 4×4, for each sample and each wavelet basis 1000 realisations of size 64×64 were simulated and then trimmed to size 40×40. Ordinary kriging has been used in the postprocessing step with a search radius of 20 and the maximum number of 16 data used. Mosaic plots for three randomly selected realisations obtained from each sample via each wavelet basis are shown in Figure 29 and the associated experimental semivariograms and quantile-quantile plots of the realisations against the corresponding samples are shown in Figures 30 and 31 respectively.

The mosaic maps of the realisations in rows 2-4 of Figure 29 indicate that the realisations from *B64n* and *B64IRn* have captured all features and spatial structure of their corresponding samples and of the data set *Berean*. For example, the location of high values in the right top corner of the two samples and the exhaustive are seen in all realisations. For sample *B64n*, the mosaic maps of the realisations simulated via both wavelet bases are consistent with the sample. Locations of low values associated

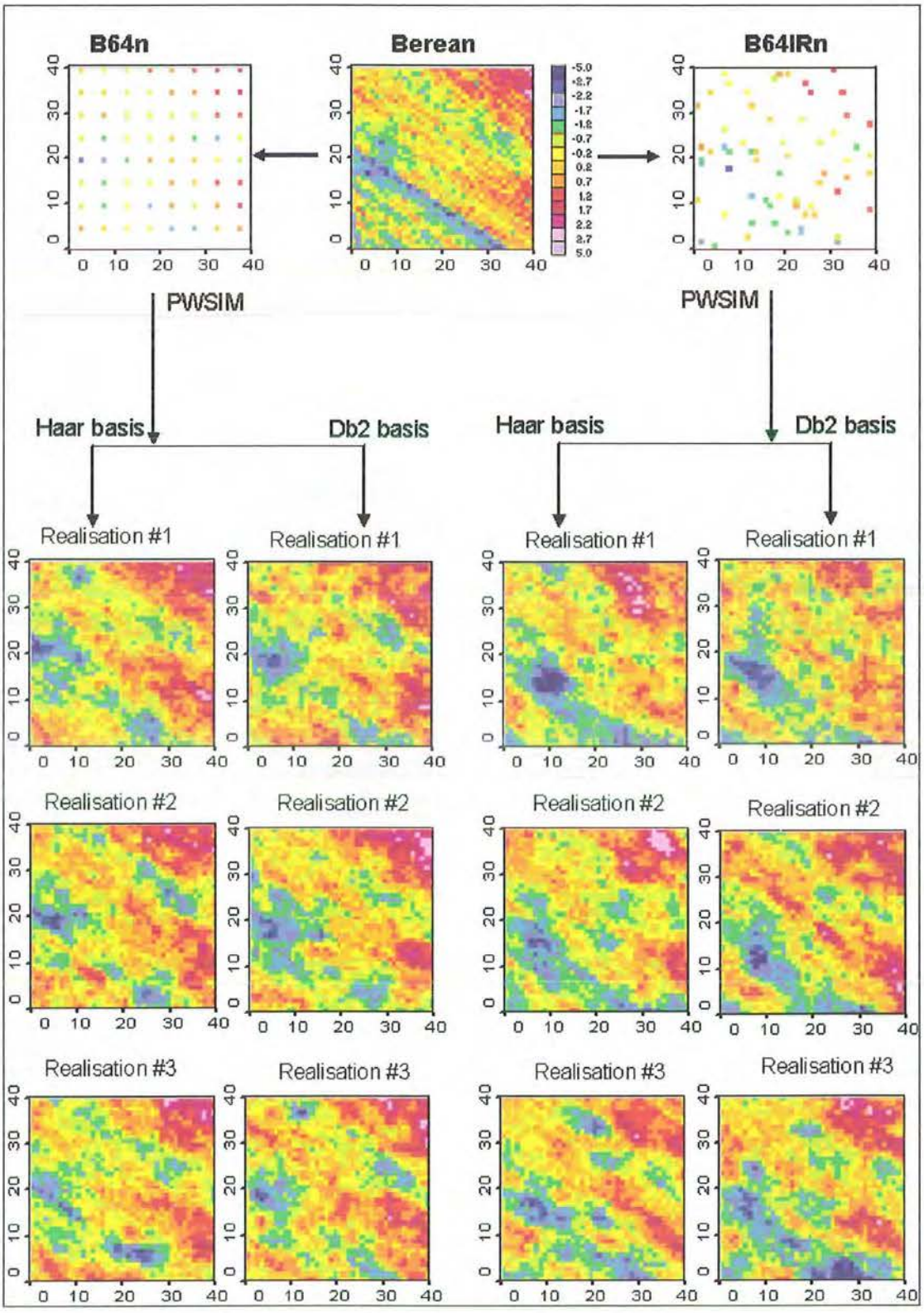


Figure 29: Mosaic plots of three randomly selected realisations from *B64n* and *B64IRn* simulated via PWSIM.

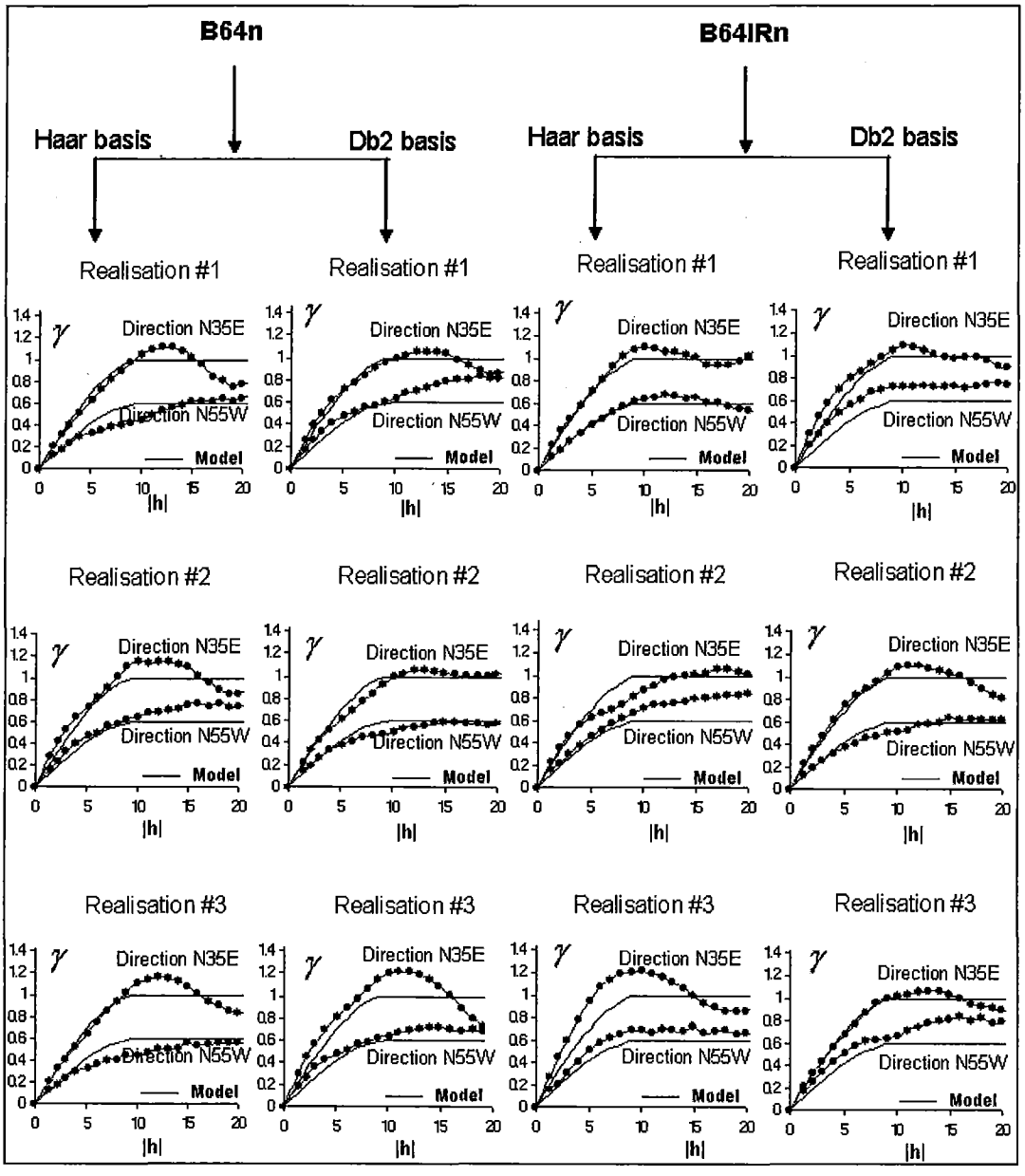


Figure 30: Semivariograms of three randomly selected realisations from *B64n* and *B64IRn* simulated via PWSIM.

Summary Statistics

	B64n							*B64IRn*						
	Sample	Realisations						Sample	Realisations					
		Haar			Db2				Haar			Db2		
Stats		1	2	3	1	2	3		1	2	3	1	2	3
Mean	0.19	0.18	0.19	0.14	0.19	0.10	0.22	-0.05	0.01	-0.03	-0.06	-0.03	0.20	0.00
Var	1.03	1.03	0.97	0.97	0.96	1.08	0.96	1.02	0.91	1.07	1.36	1.12	1.20	1.00
Max	2.36	3.06	3.00	3.06	3.64	3.40	3.20	2.15	2.76	3.76	3.14	3.17	2.91	1.81
Q3	0.91	0.87	0.87	0.78	0.87	0.71	0.90	0.64	0.70	0.61	0.71	0.63	0.95	0.69
Med	0.12	0.15	0.22	0.11	0.17	0.07	0.22	0.09	0.05	-0.07	-0.04	0.00	0.24	0.04
Q1	-0.42	-0.47	-0.47	-0.53	-0.47	-0.60	-0.46	-0.87	-0.61	-0.75	-0.86	-0.67	-0.60	-0.69
Min	-2.42	-2.96	-2.87	-2.85	-2.84	-2.76	-3.34	-2.29	-3.20	-3.46	-4.20	-4.01	-3.19	-2.93
Skew	-0.13	0.01	-0.18	0.00	-0.04	0.24	-0.02	-0.09	-0.27	0.37	-0.11	-0.15	-0.11	-0.14

Q-Q Plots

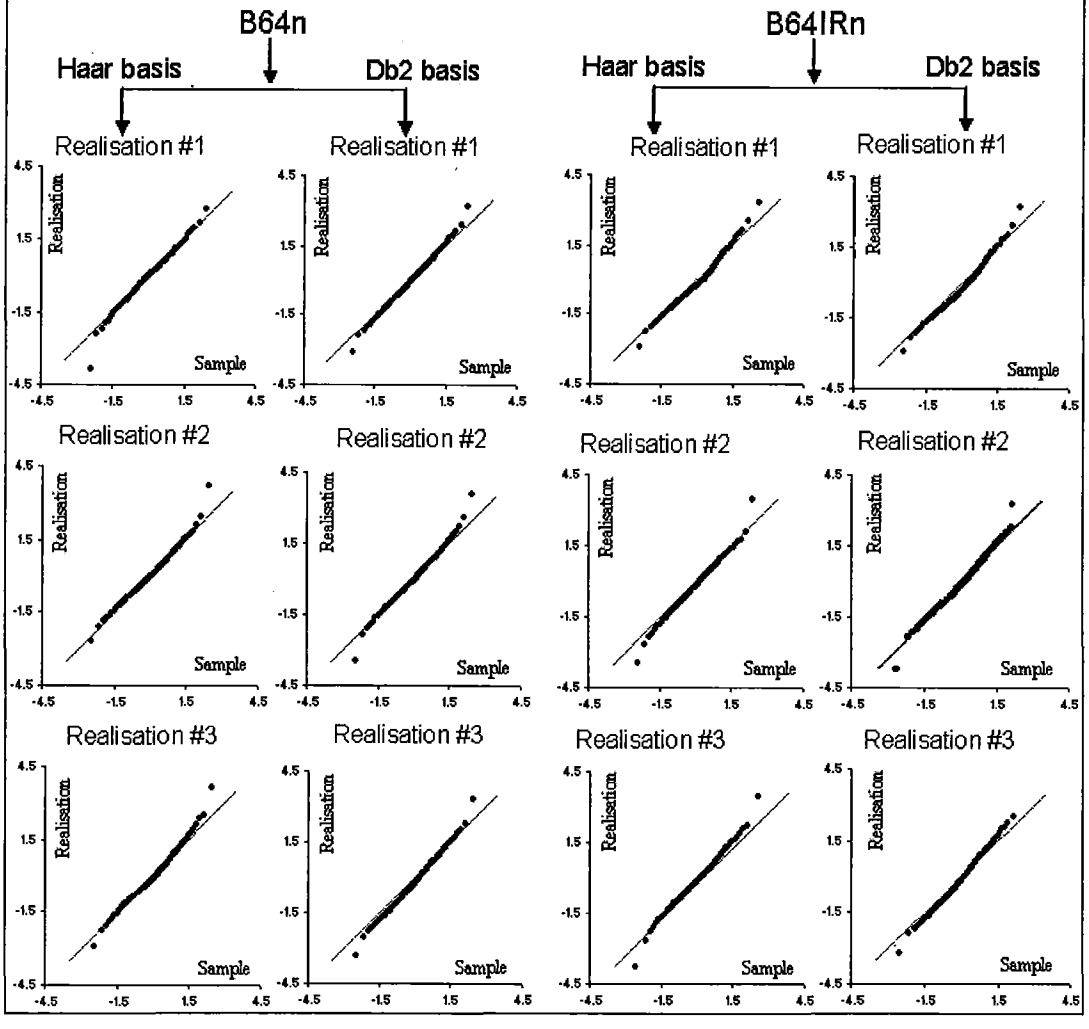


Figure 31: Summary statistics and Q-Q plots of three randomly selected realisations from *B64n* and *B64IRn* (simulated via PWSIM) against corresponding samples.

with the banding, in the right middle and close to the right bottom of the post plot of the sample are reproduced in the simulated realisations. The dark blue pixels indicate that the low extreme values in of the realisations are lower than the corresponding sample minimum and the pink pixels show that the high extreme values are higher than the corresponding sample maximum. For sample *B64IRn*, the banding in the direction $N55°W$ seen in the mosaic map of the exhaustive data set is reproduced consistently with the sample.

The experimental semivariograms of the realisations in Figure 30 show that the semivariogram models of both samples are reproduced regardless of the wavelet basis. The sill values in the direction of maximum continuity $N55°W$ and minimum continuity $N35°E$ are clearly distinguished in the semivariograms of the realisations. These semivariograms are also close to the associated semivariogram models.

The summary statistics and the quantile-quantile plots in Figure 31 indicate that the histograms of *B64n* and *B64IRn* are approximately reproduced. The means and variances of the realisations are close to the associated sample means and variances. The skewness coefficients of the realisations are also close to those of the samples, albeit with some fluctuation. Similar to the case of the isotropic samples, as a feature of the simulation, the minima of the realisations are lower than the corresponding sample minimum and the maxima of the realisations are higher than the corresponding sample maximum.

7.1.4 General Comments on PWSIM

Throughout the two case studies, we can see that PWSIM performs well in both isotropic and anisotropic cases. Regardless of the wavelet basis and the sampling method, realisations obtained via PWSIM capture the spatial structure of the asso-

ciated sample. The associated sample histograms are also reproduced. However, it should be noted that the result of the simulation depends on the kriging parameters in the postprocessing step. In addition, since PWSIM requires solving as many kriging systems as the number of unknown locations to condition the data, it is not computationally efficient.

7.2 The Direct Multi-level Wavelet-Based Conditional Simulation Algorithm DWSIM

To avoid the impact of user-defined parameters and to improve the computational efficiency of our wavelet simulation, we have proceeded to develop our direct wavelet-based conditional simulation algorithm DWSIM.

7.2.1 Description of the Algorithm DWSIM

In general, our direct multi-level wavelet-based conditional simulation algorithm DWSIM consists of two main stages, *the estimation stage* and *the back simulation stage*, each of which is carried out in a moving window.

The estimation stage uses deterministic wavelet analysis, in particular the two-dimensional discrete wavelet transform, to recursively compute conditioning values at the coarser scale $2^j, j > 0$ from the conditioning values at the original scale 2^0. The back simulation stage makes use of stochastic wavelet analysis in which the scaling coefficients are regarded as correlated random variables. As a consequence, the covariance of pairs of scaling coefficients can be used to simulate unknown scaling coefficients from the already known coefficients.

The estimation stage is based on formula (93) which indicates that the scaling coefficients are computed by means of a moving window whose size equals the size of the associated wavelet filter. This window moves on every second row and column index in ascending order on the scaling image at scale 2^j. At each stopping location, the scaling coefficient at the coarser scale 2^{j+1} associated with the window is equal to twice the weighted average of the values within the window. If the filter length is equal to two, there is no overlap in the movement of the window; if the filter length is larger than two, the succeeding window will overlap the previous one from the right

and the bottom $2M-2$ grid nodes.

By imitating the discrete wavelet transform, the estimation of the conditioning values at the coarse scale $2^{j+1}, j \geq 0$ is also carried out in a window moving in the same manner. However, since only values at sample locations are known, not all values in the window at scale 2^j are known. In many cases, the window contains no known values. Therefore, if the window contains known values, the associated scaling coefficient at the coarser level 2^{j+1} is set equal to twice the weighted average of the *known values* within the window. If the window contains no data, the associated scaling coefficient at the coarser scale is left unestimated. Since the two-dimensional discrete wavelet transform compresses the scaling image by a factor of 4, at some coarse scale, the values at all grid nodes of the associated scaling image are known. The estimation stage is completed and this scaling image is used as the *initial scaling image* for the back simulation stage.

In the back simulation stage, the scaling image at the finer scale is reconstructed from the scaling image at the next coarser scale. Since the size of the scaling image at the coarse scale 2^{j+1} equals one quarter of that of the scaling image at the finer scale 2^j, the reconstruction stage has to guarantee that the size of the scaling image at the finer scale equals four times the size of the scaling image at the coarse scale.

In order to do so, DWSIM makes use of the scaling coefficients and the covariance of the scaling coefficients at the same scale as well as at two consecutive scales. The expansion is also carried out by means of non-overlapping windows of size 2×2. In the expansion, each unknown value in the window moving on the scaling image at the finer scale 2^j is expressed as a linear combination of the scaling coefficients at the coarser scale 2^{j+1} plus a normally distributed random noise. The weights of the linear combination and the standard deviation of the noise are determined in such a way

that the correlation among the scaling coefficients at the coarse scale 2^{j+1} and the correlation between the scaling coefficients at scale 2^j and scale 2^{j+1} are maintained. The wavelet coefficients are ignored since they contain very little information of the scaling image at the previous scale.

The estimation and back simulation stages are detailed in the following Sections in which the method of obtaining the conditioning values at the coarse scales is applied for Haar wavelets and then generalised to Daubechies wavelets with M vanishing moments. The back simulation stage is the same for all wavelet bases.

Estimation Stage for Haar Wavelets

Let $\mathcal{A} \subseteq \mathbb{R}^2$ be the study region of size $P \times Q$ where $P = 2^{m_1}$ and $Q = 2^{n_1}$ with $m_1, n_1 = 1, 2, 3...$ then at scale 2^{j+1}, the size of the scaling image is $(P/2^{j+1}) \times (Q/2^{j+1})$ where $j = 0, 1, ..., \min(m_1, n_1)$. The estimation stage is preceded by computing the covariance tables for the coarser scales $2^j, j > 0$. The computational effort for the computation of the covariance tables will be discussed in Appendix 12.3.

For Haar wavelets, since the filters have only two non-zero coefficients, the estimation of the conditioning values at scale $2^{j+1}, j \geq 0$ is carried out by means of non-overlapping windows of size 2×2. Each window contains four values denoted by $c^j[2m + u, 2n + v]$ where $u, v = 0, 1$ and $0 \leq m < P/2^{j+1}, 0 \leq n < Q/2^{j+1}$.

According to formula (93) and since for Haar wavelets the values of the lowpass filter h are defined by

$$h[n] = \begin{cases} \sqrt{2}/2 & n = 0, 1 \\ 0 & otherwise \end{cases}$$

the scaling coefficient $c^{j+1}[m, n]$ at the coarser scale 2^{j+1} is equal to twice the average of the scaling coefficients $c^j[2m + u, 2n + v], u, v = 0, 1$ at the finer scale. For this

reason, each scaling coefficient at the coarser scale is set equal to twice the average of the known values within the associated window at the previous finer scale. When moving from the original scale to the next coarser scale, a random noise, drawn from a standard normal distribution and independent of (m, n), is added to the estimated scaling coefficient $c^1[.,.]$. The purpose of adding this noise is to avoid uniqueness in the estimation in order to improve the variability of the simulation. Once all scaling coefficients associated with the windows that contain conditioning data have been estimated, the variance of the newly estimated conditioning values is computed. If this variance is different from the variance of the associated scaling image (read from the precomputed covariance table), the estimated conditioning values for each coarse scale are re-scaled by a factor f, where

$$f = \sqrt{\frac{\text{variance of the associated scaling image}}{\text{variance of the estimated conditioning values}}}, \tag{138}$$

so that their variance is equal to the variance of the associated scaling image before moving to the next coarser scale.

The process is illustrated in Figure 32 for the case of an original image of size 8×8. In Figure 32A1 the sample locations are indicated by dots and the windows that contain conditioning values are shaded. After the first estimation, four locations that have estimated conditioning values at scale 2^1 are shown in Figure 32A2. Then the windows that cover these locations are indicated by the shaded areas in Figure 32B1. After the two-dimensional discrete wavelet transform has been applied one more time, the image with all of grid locations estimated is obtained as shown in Figure 32B2. In the case of Haar wavelets, since the filter length is equal to two and the dimensions of the study regions are a power of two, there is no edge effect. The estimated scaling image in which the values at all grid nodes are known together with all intermediate conditioning values obtained in the estimation stage are used

to simulate the original image in the back simulation stage.

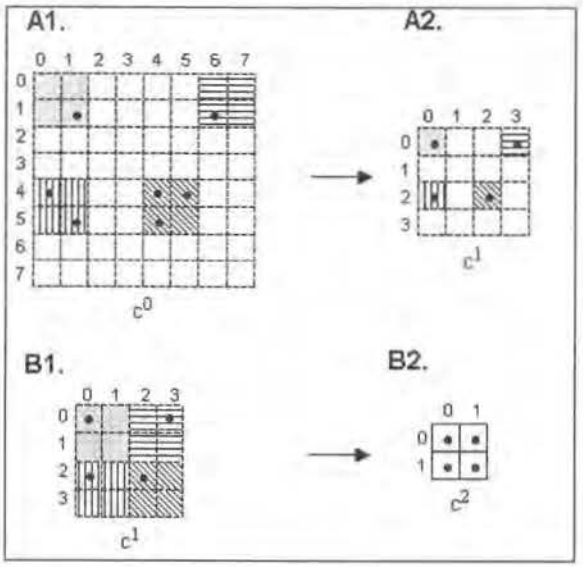


Figure 32: Illustration of estimation stage in the Haar basis case.

Estimation Stage for Daubechies wavelets

For Daubechies wavelets with $2M$ non-zero coefficients, the estimation stage is carried out by means of overlapping windows of size $(2M) \times (2M)$. A window, whose values are denoted by $c^j[2m+u, 2n+v]$ where $u, v = 0, 1, ..., 2M-1$ and $0 \leq m < P/2^{j+1}, 0 \leq n < Q/2^{j+1}$, starts only at even indices and moves across the finer scaling image in row and column ascending order to search for data. At each stopping location, the scaling coefficient $c^{j+1}[m,n]$ associated with the window is estimated by twice the weighted average of the known values in the window. A random noise drawn from a standard normal distribution is added to the estimated scaling coefficient $c^1[m,n]$ when moving from the original scale to the next coarser scale to improve the variability of the simulation. In other words, let

$$h'[u]h'[v] = \begin{cases} h[u]h[v] & \text{if} \quad c^j[2m+u, 2n+v] \text{ is known} \\ 0 & \text{if} \quad c^j[2m+u, 2n+v] \text{ is unknown} \end{cases}$$

then the scaling coefficient associated with a window containing conditioning value(s) is estimated by

$$c^{j+1}[m,n] = \frac{\sum_{u=0}^{2M-1}\sum_{v=0}^{2M-1} h'[u]h'[v]c^{j}[2m+u, 2n+v]}{\sum_{u=0}^{2M-1}\sum_{v=0}^{2M-1} h'[u]h'[v]} \times 2 + R \qquad (139)$$

where R is drawn from $N(0,1)$ if $j=0$ and $R=0.0$ if otherwise and is independent of $[m,n]$. Due to the negative values of the lowpass filter coefficients, the sum

$$\sum_{u=0}^{2M-1}\sum_{v=0}^{2M-1} h'[u]h'[v]$$

in equation (139) can take values very close to or equal to zero. In the case where the absolute value of the denominator in (139) is smaller than a predefined tolerance, the estimation in the associated window is modified by

1. Computing the average of the known values in the window.
2. Assigning the average to the locations in the window whose values are unknown.
3. Computing the scaling coefficient $c^{j+1}[m,n]$ using formula (93) then adding a random noise drawn from $N(0,1)$ if $j=0$.

Theoretically, the estimation can be applied to Daubechies wavelets with M vanishing moments. However, the larger the size of the window, the less accurate the above estimation becomes. In addition, as will be shown in Chapter 8, the higher the number of vanishing moments, the more computation is required in the simulation. Therefore, only the Haar and the Db2 wavelet bases are used in DWSIM.

Similar to the case of the Haar wavelet basis, the estimated conditioning values at each coarse scale are also rescaled so that their variance is equal to the variance of associated scaling coefficients.

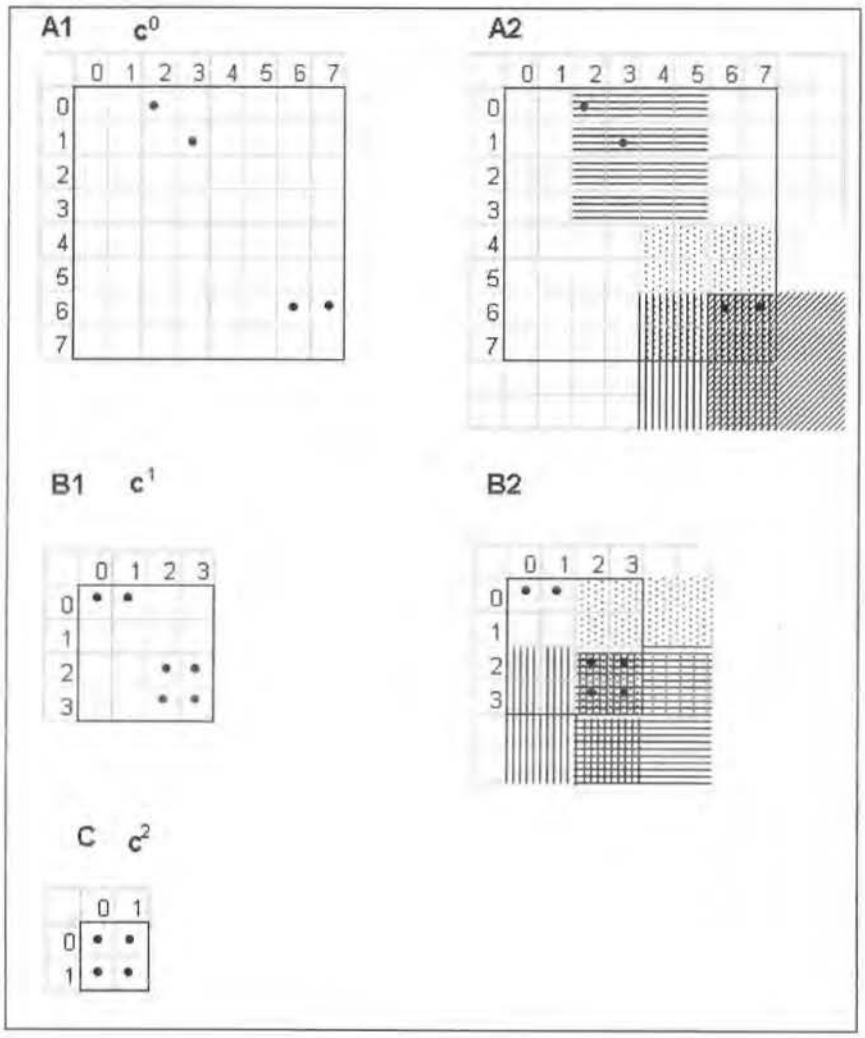


Figure 33: Illustration of estimation stage in the case of Db2 wavelets.

The estimation of the scaling coefficients used as conditioning values at scale $2^j, j > 0$ for the Db2 wavelet basis is illustrated in Figure 33 for an original image of size 8×8. Unlike the case of the Haar basis, in the estimation of scaling coefficients at scale $2^j, j > 0$, for the Daubechies wavelet basis one has to take into account the edge effect. At the right and bottom boundaries, the scaling image at the finer scale

has to be extended two grid nodes further so that the window size is always equal to the filter length. In Figure 33A1 sample locations at the original level $j = 0$ are indicated by dots and the windows that contain conditioning values are shaded in Figure 33A2. The locations of the estimated conditioning values after the first application of the two-dimensional discrete wavelet transform and the associated windows that cover these locations are shown in Figure 33B1 and Figure 33B2, respectively. After applying the two-dimensional discrete wavelet transform one more time, a scaling image with all grid locations estimated is obtained as shown in Figure 33C.

The Back Simulation Stage

In the back simulation stage, the scaling image at scale $2^j, j \geq 0$ is simulated by expanding the scaling image at the coarser scale 2^{j+1} by a factor of 4. The expansion makes use of the covariance of scaling coefficients within the same scale as well as the covariance of scaling coefficients across two consecutive scales. It is carried out by means of non-overlapping windows of size 2×2 in which each unknown value is simulated in terms of the known scaling coefficients at the next coarser scale. The following Sections provide the formula for the expansion and the computation of the related elements used in the expansion.

The expansion of the scaling image from one scale to the next finer scale is accomplished by expressing each unknown scaling coefficient in the moving window as a linear combination of scaling coefficients at the coarser level plus a random noise. The basis for the formula lies in the fact that, the scaling coefficient $c^{j+1}[m,n]$ at scale 2^{j+1} is a linear combination of the scaling coefficients $c^j[2m+u, 2n+v]$, where $u,v = 1,2,...,2M$ and M is the number of vanishing moments. Since the coefficients c^j at scale 2^j are correlated, the coefficients c^{j+1} are also correlated. The correlation

is not confined only to the coefficients in the same scale, it also occurs among scaling coefficients at different scales. Because of the correlation, a scaling coefficient at one scale can be simulated from the coefficients at another scale.

To expand the whole scaling image at a coarser scale to the previous finer scale by a factor of four, each scaling coefficient of the scaling image $\{c^{j+1}\}$ has to be expanded into four scaling coefficients of the previous finer scale 2^j. To optimise the expansion, for a given scaling coefficient $c^{j+1}[m,n]$ at location (m,n), it is necessary to identify four scaling coefficients at the finer scale 2^j that have the strongest correlation with $c^{j+1}[m,n]$. Then these four scaling coefficients are simulated in terms of $c^{j+1}[m,n]$.

Proposition 7 *For a given location (m,n) in the scaling image at scale 2^{j+1}, there exists an associated window of size 2 by 2 in the scaling image at scale 2^j that contains four scaling coefficients denoted by $c^j[2m+u,2n+v]$ where $u,v=0,1$ and $0 \le m < P/2^{j+1}, 0 \le n < Q/2^{j+1}$ that have the strongest correlation with $c^{j+1}[m,n]$.*

Proof. For Haar wavelets, according to formula (93), only the four values $c^j[2m+u,2n+v], u,v=0,1$ are used to compute $c^{j+1}[m,n]$. Therefore the correlation between $c^{j+1}[m,n]$ and each scaling coefficient in the 2×2 window $(2m+u,2n+v), u,v=0,1$ is stronger than the correlation between $c^{j+1}[m,n]$ and any scaling coefficient that lies outside the window.

For Db2 wavelets, using formula (93), the weights $h[u]h[v]$ associated with $c^j[2m+u,2n+v], u,v=0,1,2,3$ used to compute $c^{j+1}[m,n]$ in the linear combination are shown in Table 4.

Since the weights $h[u]h[v]$ associated with four values $c^j[2m+u,2n+v], u,v=0,1$, defined by

$$h[0]h[0] = .2333,$$

u\v	0	1	2	3
0	0.2333	0.4040	0.1083	−0.0625
1	0.4040	0.6998	0.1875	−0.1083
2	0.1083	0.1875	0.0502	−0.0290
3	−0.0625	−0.1083	−0.0290	0.0167

Table 4: Weights used to compute a scaling coefficient from scaling coefficients at previous finer scale.

$$h[0]h[1] = .4040,$$

$$h[1]h[0] = .4040,$$

$$h[1]h[1] = .6998,$$

respectively, are larger than the rest of the weights, the scaling coefficients $c^j[2m + u, 2n + v], u, v = 0, 1$ have stronger correlation with $c^{j+1}[m, n]$ compared to other scaling coefficients at scale 2^j. ■

Because the correlation between $c^{j+1}[m, n]$ and $c^j[2m + u, 2n + v], u, v = 0, 1$, is stronger than the correlation between $c^{j+1}[m, n]$ and any other scaling coefficient outside the window, it is more efficient to simulate these scaling coefficients from the scaling coefficient $c^{j+1}[m, n]$ than to simulate any other scaling coefficients outside the window from $c^{j+1}[m, n]$. In addition, to take into account the correlation between the scaling coefficients c^j and the scaling coefficients c^{j+1} in the neighbourhood centred at $c^{j+1}[m, n]$, we propose a formula for simulating each of the scaling coefficient c^j in

the window by putting

$$c^j[2m+u, 2n+v] = \sum_{k_2,l_2=-i}^{i} \alpha_{k_2,l_2}^{j+1}[u,v] c^{j+1}[m+k_2, n+l_2] + \gamma^{j+1}[u,v] R_{m,n}[u,v]. \tag{140}$$

In (140) $R_{m,n}[u,v]$ is a random number drawn from a standard normal distribution and by construction is uncorrelated with c^{j+1}; α_{k_2,l_2}^{j+1} are called the weights; $\gamma^{j+1}[u,v]$ is called the standard deviation of the noise and the index $i \geq 1$ in the summation is called the radius of the estimation. It can be seen from (140) that each c^j is a linear combination of $(2i+1)^2$ scaling coefficients centered at $c^{j+1}[m,n]$.

Unlike in the case of WSIM, where wavelet coefficients are computed in terms of the scaling coefficients at the same scale, here we compute the scaling coefficients directly in terms of the scaling coefficients at the coarser scale and so avoid the use of wavelet coefficients completely. By properly defining the weights and the standard deviation of the noise, simulated realisations will reproduce the global statistics of the sample.

Defining the Value of the Estimation Radius From equation (140) it can be seen that the number of scaling coefficients centred at $c^{j+1}[m,n]$ is determined by the estimation radius i. Hence, it is necessary to define an appropriate value for i to avoid unnecessary computation and to ensure that no important information is ignored. According to Vannucci and Corradi (1997) the covariance between two scaling coefficients at two consecutive scales $c^{j+1}[p,q]$ and $c^j[k,l]$ vanishes when

$$|k-2p| > 2M-1 \text{ or } |l-2q| > 2M-1 \tag{141}$$

where M is the number of vanishing moments.

In our case, we focus on the covariance between the scaling coefficients in the

window $c^j[2m+u, 2n+v]$ and the scaling coefficients $c^{j+1}[m \pm i, n \pm i]$ centered at $c^{j+1}[m, n]$.

By letting $k = 2m+u, p = m \pm i, l = 2n+v$ and $q = n \pm i$, we have from (141)

$$|2m+u-2(m \pm i)| > 2M-1 \text{ or } |2n+v-2(n \pm i)| > 2M-1. \tag{142}$$

Since $u, v = 0, 1$, the left inequation in (142) gives

$$\begin{aligned} u+2i &> 2M-1 \text{ or } u-2i < 1-2M \\ i &> M - \frac{1+u}{2} \text{ or } i > M - \frac{1-u}{2}. \end{aligned} \tag{143}$$

Similarly, the right inequation in (142) can be simplified as

$$i > M - \frac{1+v}{2} \text{ or } i > M - \frac{1-v}{2}. \tag{144}$$

From (143) and (144) for all u, v the correlation between $c^j[2m+u, 2n+v]$ and $c^{j+1}[m \pm i, n \pm i]$ vanishes when $i > M$. Therefore, only the scaling coefficients $c^{j+1}[m \pm M, n \pm M]$ need to be included in equation (140). That is, i is equal to M. This choice of i was also verified using trial and error in the case of the Haar wavelets in Tran et al (2002a).

Since the covariance between two grid nodes of the scaling image at scale 2^{j+1} is read from a precomputed covariance table, to ensure that the value is listed in the table, the index i must also not exceed $S_m/2$ where

$$S_m = \min(P/2^{j+1}, Q/2^{j+1}) \text{ for } j \geq 0$$

Computation of the Weights The process of computing the weights $\alpha^{j+1}_{k_2,l_2}$ in this part will ensure that the simulated realisations at the original level preserve the spatial correlation of the initial scaling image estimated from the sample.

In order to determine the weights we first multiply equation (140) by $c^{j+1}[m+k_1, n+l_1]$, where $k_1, l_1 = -i, ..., 0, ..., i$, to have a system of $(2i+1)^2$ equations

$$\begin{aligned} & c^j[2m+u, 2n+v]c^{j+1}[m+k_1, n+l_1] \\ = \; & c^{j+1}[m+k_1, n+l_1] \sum_{k_2,l_2=-i}^{i} \alpha_{k_2,l_2}^{j+1}[u,v]c^{j+1}[m+k_2, n+l_2] \\ & +\gamma^{j+1}[u,v]c^{j+1}[m+k_1, n+l_1]R_{m,n}[u,v] \end{aligned}$$

The correlation of the scaling coefficients c^{j+1} with the coefficients to be simulated $c^j[2m+u, 2n+v]$ is taken into account by taking the expected values for both sides of the above equations. Hence, the covariances between the scaling coefficient to be simulated at the finer scale 2^j and the scaling coefficients at the coarse scale 2^{j+1} are related by

$$\begin{aligned} & \delta^j_{(m+k_1,2m+u),(n+l_1,2n+v)} \\ = \; & E\left\{ \sum_{k_2,l_2=-i}^{i} \alpha_{k_2,l_2}^{j+1}[u,v]c^{j+1}[m+k_1, n+l_1]c^{j+1}[m+k_2, n+l_2] \right\} \\ & +E\left\{\gamma^{j+1}[u,v]c^{j+1}[m+k_1, n+l_1]R_{m,n}[u,v]\right\} \end{aligned} \tag{145}$$

Since $R_{m,n}[u,v]$ and $c^{j+1}[.,.]$ are uncorrelated, the second term on the right hand side of system (145) equals zero. Therefore, from (145) we have a system of $(2i+1)^2$ linear equations

$$\delta^j_{(m+k_1,2m+u),(n+l_1,2n+v)} = \sum_{k_2,l_2=-i}^{i} \alpha_{k_2,l_2}^{j+1}[u,v]\beta^{j+1}_{(m+k_2,m+k_1),(n+l_2,n+l_1)} \tag{146}$$

where $\beta^{j+1}_{(m+k_2,m+k_1),(n+l_2,n+l_1)}$ denotes the covariance of scaling coefficients at scale 2^{j+1}. Thus the weights $\alpha_{k_2,l_2}^{j+1}[u,v]$ are the solution of system (146).

Using matrix notation, system (146) can be written as

$$B\alpha = \Delta \tag{147}$$

with

$$B = \left[\beta^{j+1}_{(m+k_2,m+k_1),(n+l_2,n+l_1)}\right]_{i,k_2,l_2,k_1,l_1=-i,\ldots,0,\ldots,i}$$

$$\alpha = \left[\alpha^{j+1}_{k_2,l_2}[u,v]\right]^T$$

where T denotes the matrix transpose, and Δ is the vector

$$\Delta = \left[\delta^j_{(m+k_1,2m+u),(n+l_1,2n+v)}\right]$$

Computation of the Standard Deviation The standard deviation of the simulated scaling image at scale $2^j, j \geq 0$ is reproduced by suitably defining the standard deviation $\gamma^{j+1}[u,v]$ of the random noise. This standard deviation is computed by calculating the standard deviation of the simulated values in the window $c^j[2m+u, 2n+v]$. By squaring both sides of (140), we have

$$\left[c^j[2m+u,2n+v]\right]^2 = \left[\sum_{k_2,l_2=-i}^{i} \alpha^{j+1}_{k_2,l_2}[u,v]c^{j+1}[m+k_2,n+l_2] + \gamma^{j+1}[u,v]R_{m,n}[u,v]\right]^2 \tag{148}$$

and as $R_{m,n}[u,v]$ and c^{j+1} are uncorrelated, by taking the expected values for both sides equation (148) becomes

$$\begin{aligned} E\left\{c^j[2m+u,2n+v]\right\}^2 &= E\left\{\sum_{k_2,l_2=-i}^{i} \alpha^{j+1}_{k_2,l_2}[u,v]c^{j+1}[m+k_2,n+l_2]\right\}^2 \\ &\quad + E\left\{\gamma^{j+1}[u,v]R_{m,n}[u,v]\right\}^2 \end{aligned} \tag{149}$$

Since $R_{m,n}[u,v]$ are drawn from the standard normal distribution

$$E\left\{R[u,v]\right\}^2 = 1$$

and

$$E\left\{c^j[2m+u,2n+v]\right\}^2 = \beta^j_{(2m+u,2m+u),(2n+v,2n+v)}$$

we have that

$$\beta^j_{(2m+u,2m+u),(2n+v,2n+v)} = \sum_{k_2,l_2=-i}^{i}\sum_{k_1,l_1=-i}^{i} \alpha^{j+1}_{k_2,l_2}[u,v]\alpha^{j+1}_{k_1,l_1}[u,v]\beta^{j+1}_{(m+k_2,m+k_1),(n+l_2,n+l_1)} + \left[\gamma^{j+1}[u,v]\right]^2 \quad (150)$$

The value $\beta_{(.)}$ in the left hand side of equation (150) is the standard deviation of the scaling coefficients at scale 2^j.

In matrix form, equation (150) is rewritten as

$$\beta^j_{(2m+u,2m+u),(2n+v,2n+v)} = \alpha^T B\alpha + \left[\gamma^{j+1}[u,v]\right]^2$$

Hence the standard deviation $\gamma^{j+1}[u,v]$ is determined by

$$\gamma^{j+1}[u,v] = \sqrt{\beta^j_{(2m+u,2m+u),(2n+v,2n+v)} - \alpha^T B\alpha}. \quad (151)$$

Because the configuration of locations of scaling coefficients on the left hand side of (140) is independent of location (m,n) (see Proposition 5), the covariance matrix B in (147) and (151) is also independent of (m,n). In addition, since the configuration between the window and the locations of scaling coefficients on the right hand side of (140) does not depend on (m,n), the vector Δ on the right hand side of (147) also does not depend on (m,n). Therefore, the weights $\alpha^{j+1}_{k_2,l_2}[u,v]$ and the value $\gamma^{j+1}[u,v]$ do not depend on the location of the window but only on (u,v), the location in the window of the scaling coefficient to be simulated. Therefore, for each pair (u,v), system (147) and equation (151) need to be solved once. In other words, to move from a coarse level to the previous finer level, system (147) and equation (151) are solved a total of four times.

Since the value of index i depends on the number of vanishing moments, the higher the number of vanishing moments the finitely supported mother wavelet basis has,

the larger the system (147) is. The system (147) contains nine linear equations with nine unknowns if the basis used is the Haar wavelets and $(2M+1)^2$ equations with $(2M+1)^2$ unknowns for DbM wavelets. The larger the system, the more computation is required. Therefore, to save computation time, we limit the bases in DWSIM only to the Haar wavelets and the Db2 wavelets.

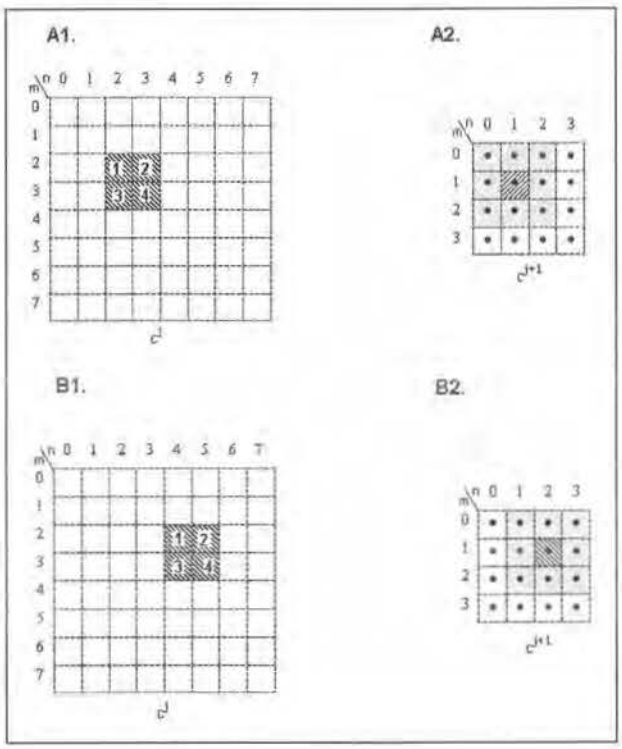


Figure 34: A1 &B1/ Scaling images with windows containing values to be simulated; A2 & B2: locations of scaling coefficients used to simulate values in associated window.

Figure 34 gives an illustration of the simulation of four coefficients $c^j[2m+u, 2n+v]$ in the case $(m,n) = (1,1)$ and $(m,n) = (1,2)$. Figures 34A1 and 34B1 show locations of the windows that contain the coefficient c^j to be simulated (shaded regions). Locations of c^j in the window associated with $(u,v) = (0,0); (u,v) = (0,1); (u,v) = (1,0)$ and $(u,v) = (1,1)$ are denoted by $1, 2, 3$ and 4, respectively. Figures 34A2 and 34B2 show locations of the associated coefficients c^{j+1} in the case $i = 1$. The elements of

matrix B in equation (147) are the covariance of the scaling coefficients whose locations are shaded. It is obvious that matrix B does not depend on (m, n); however, the covariances between coefficients c^j in the shaded windows in Figures 34A1 and 34B1 and the coefficients c^{j+1} in the shaded areas in Figures 34A2 and 34B2 are functions of (u, v).

When the window comes close to the boundary, in many cases some scaling coefficients in the coarser scale lie outside the scaling image. In this situation, the edge problem is handled by setting the associated weights equal to zero. Figures 35A2 and 35B2 show the coefficients used in the linear combinations in equation (140) when $(m, n) = (0, 0)$ and $(m, n) = (1, 3)$ in the case $i = 1$ for a coarser scaling image of size 4×4.

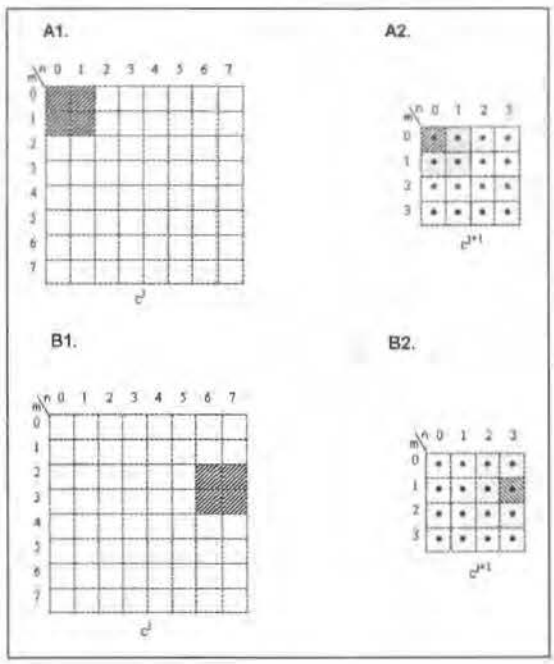


Figure 35: A1&B1/ Scaling images and windows containing values to be simulated. A2&B2/ Locations of scaling coefficients (shaded) used to simulate values in windows in A1& B1.

Summary of the Algorithm DWSIM After fitting a semivariogram model to the experimental semivariogram of the data and computing the covariance model from the semivariogram model using $C(\mathbf{h}) = 1 - g(\mathbf{h})$, the DWSIM algorithm is carried out by the following steps:

1. For $j = 0$. Compute the covariance table for the covariance model.

2. Compute the covariance table for scale 2^{j+1}. Compute the coefficients (the weights and the standard deviation) in (140) associated with the scale. Store the coefficients.

3. Extend the scaling image at scale 2^j to the right and to the bottom $(2M - 2)$ grid nodes.

4. Construct a window of size $(2M) \times (2M)$ whose first row and first column overlap the first row and the first column of the scaling image. If the window contains conditioning value(s), estimate the associated conditioning value for the scale 2^{j+1} and store this value; if not move to the next window location, defined at every second row and column index in row-ascending order. The process is continued until all grid node values of the scaling image at scale 2^{j+1} associated with windows that contain known value(s) at scale 2^j are estimated.

5. Rescale the estimated values so that the variance of the estimated values equal to the target variance.

6. If all grid nodes at scale 2^{j+1} are estimated, the scaling image at this scale is the initial scaling image, if NOT let $j = j + 1$ and go back to step (2).

7. Let $j = j - 1$.

8. In the scaling image at scale 2^j, construct a window of size 2×2 whose first row and first column overlap the first row and the first column of the scaling image.

9. Calculate the unknown values in the window in terms of scaling coefficients at scale 2^{j+1} via formula (140) using the weights computed in step 2.

10. Move to the next window location and repeat step (9) until the values at all grid nodes in this scaling image is simulated.

11. Repeat step (7)-(10) until $j < 0$.

In the algorithm, the estimation stage includes steps (1)-(6), and the back simulation stage consists of steps (7)-(11). The weights used in the back simulation stage are computed in step (2) to avoid storing the covariance tables.

In the case where the sample is not standard normal, the data have to be transformed into normal scores before the modelling of the spatial continuity. The results of the simulation then have to be back-transformed into the attribute values. The summary of the algorithm DWSIM is shown in Figure 36.

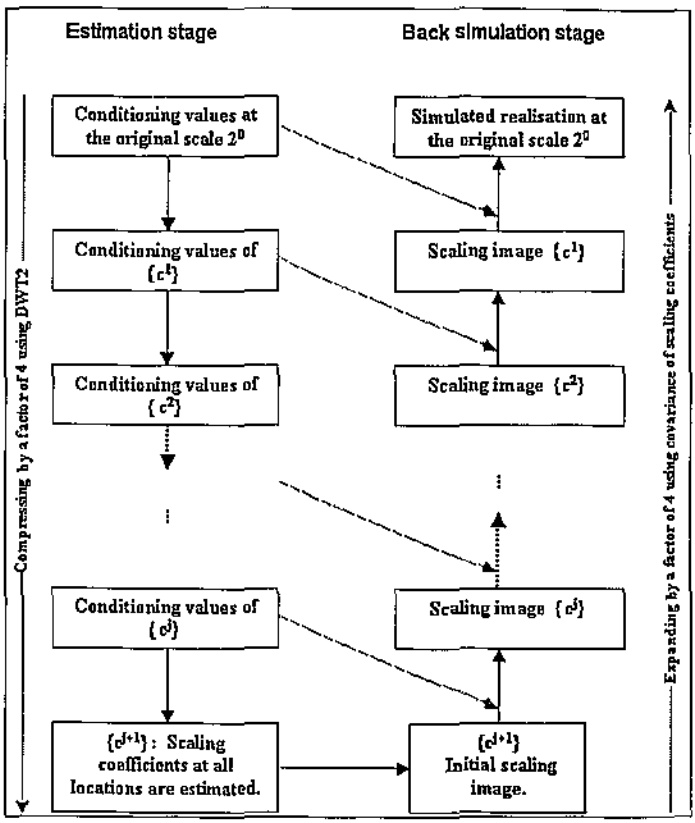


Figure 36: Summary of DWSIM.

7.2.2 Case Study 3: Applications of DWSIM to Isotropic Samples

The isotropic samples used to test the performance of DWSIM are *pH100n* and *pH100IRn*. As DWSIM requires the dimensions of the study region to be power of 2, the study region was extended to size 64×64 and after the conclusion of the simulation, the simulated realisations were trimmed to 61×61. For each wavelet basis, 1000 realisations from each sample were simulated. The mosaic maps of three randomly selected realisations for each case were plotted to compare with the mosaic plot of the exhaustive data set *pHsoiln* and the post plots of the associated samples. Sample histogram reproduction was visualised via the quantile-quantile plots of the realisations against the associated samples. Spatial continuity reproduction was visualised via the plots of the experimental semivariograms of the simulated realisations together with the associated semivariogram models. The mosaic plots of the realisations from *pH100n* and *pH100IRn* are shown in Figure 37 while the associated semivariograms and the quantile-quantile plots are shown in Figures 38 and 39 respectively.

The mosaic plots for the realisations in rows 2, 3 and 4 of Figure 37 indicate that the simulated realisations have captured all features of the samples, regardless of the wavelet basis and of the sampling method. No artefacts have been detected in these mosaic plots. Locations with high and low values in the data set *pHsoiln* and the samples are reproduced in the simulated realisations.

The spatial continuity is also reproduced since all of the experimental semivariograms in Figure 38 are similar to the associated semivariogram models. The summary statistics and the quantile-quantile plots in Figure 39 show that the histograms of both samples are approximately reproduced. The means, variances and quartiles of the simulated realisations are close to those of the samples with the variances of all realisations from *B64n* simulated using the Haar basis being lower than the

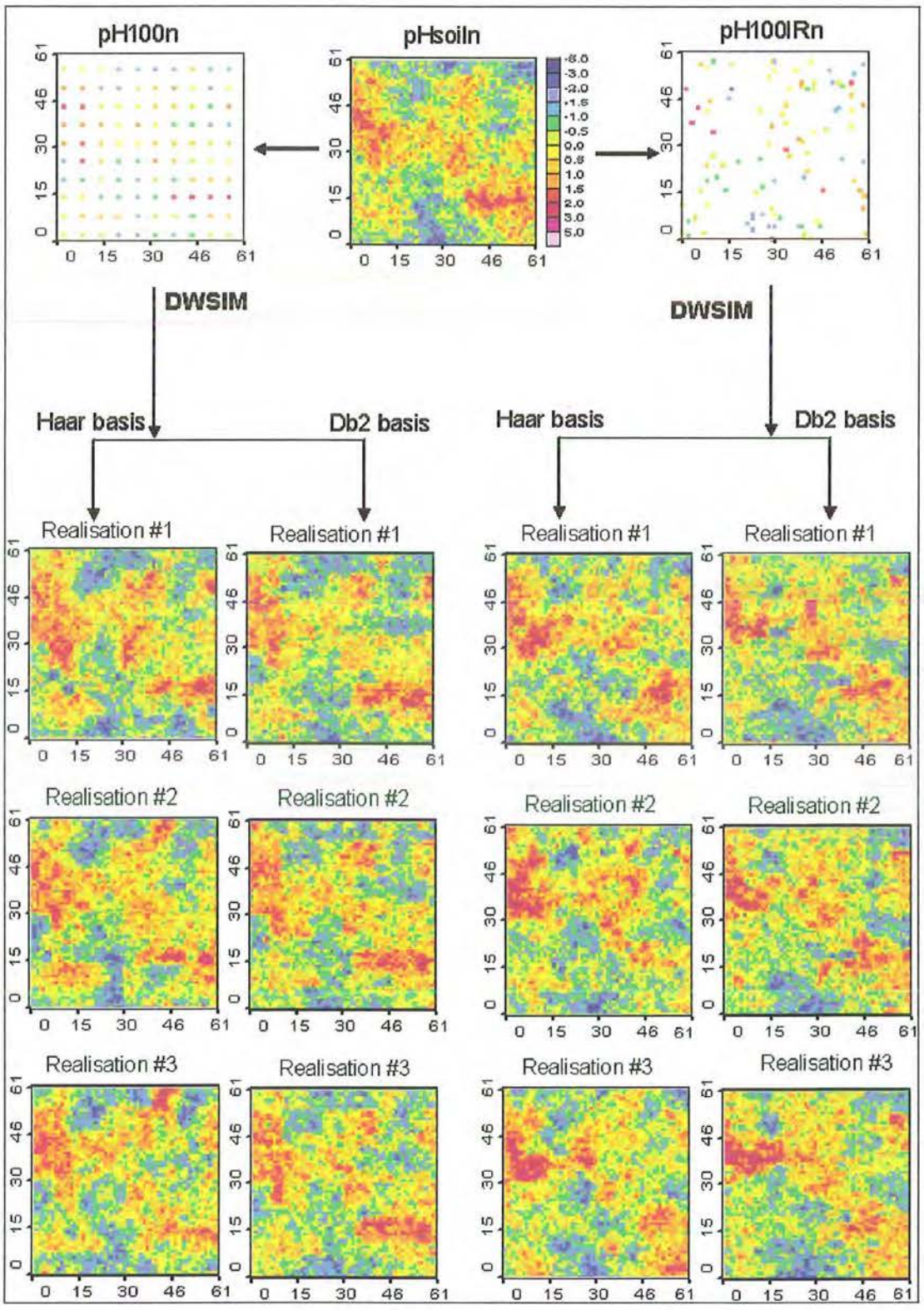


Figure 37: Mosaic plots of three randomly selected realisations from *pH100n* and *pH100IRn* simulated by DWSIM.

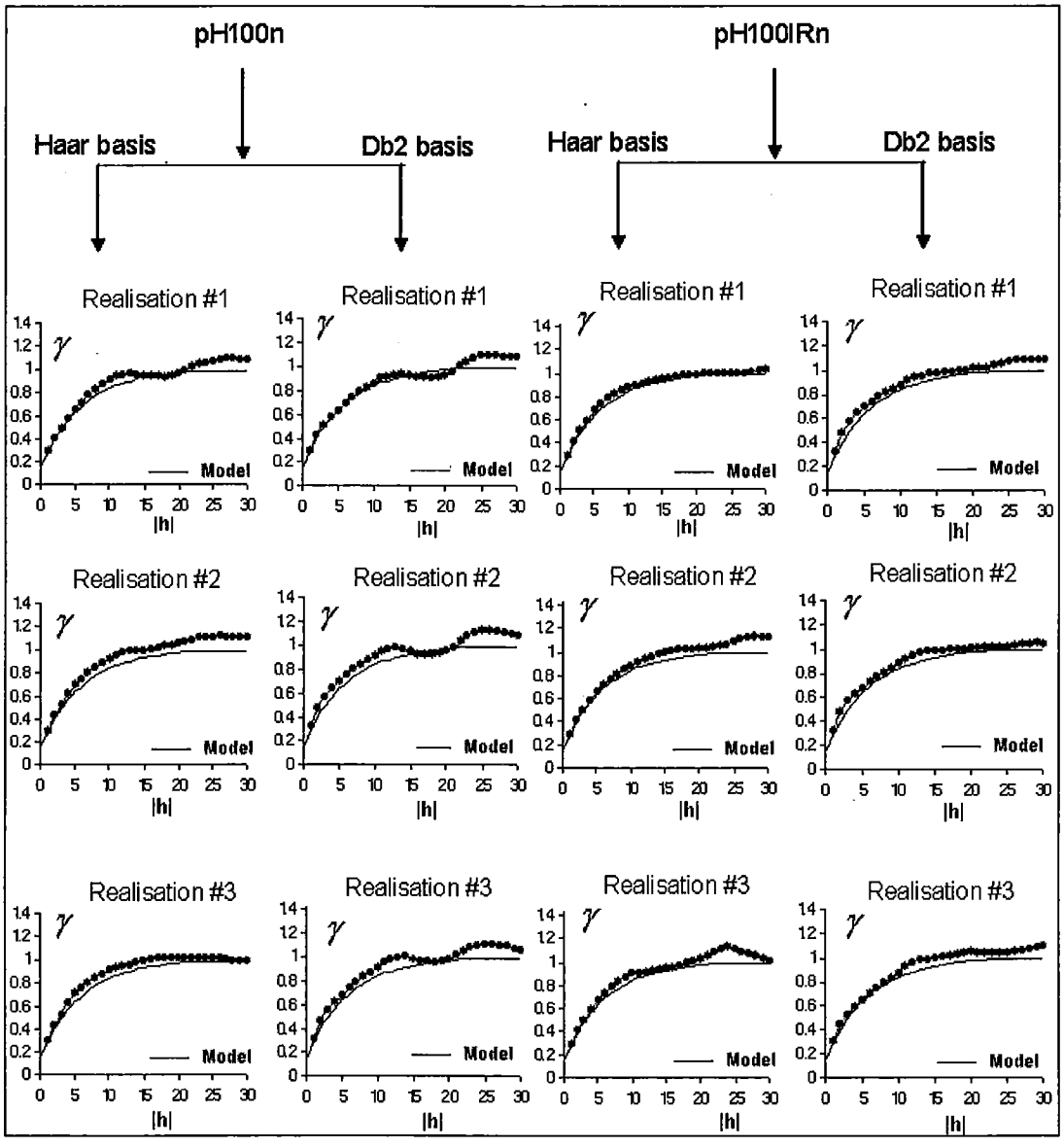


Figure 38: Semivariograms of the three randomly selected realisations from *pH100n* and *pH100IRn* simulated by DWSIM.

Summary Statistics

	pH100n							*pH100IRn*						
	Sample	Realisations						Sample	Realisations					
		Haar			Db2				Haar			Db2		
Stats		1	2	3	1	2	3		1	2	3	1	2	3
Mean	0.04	0.04	0.04	-0.04	0.02	0.02	0.01	-0.12	0.03	-0.02	0.01	0.00	0.07	0.01
Var	1.02	1.05	1.02	1.04	0.98	0.96	0.98	1.00	1.04	1.05	1.05	1.01	1.06	1.03
Max	2.54	3.15	3.86	3.30	3.19	3.17	3.68	2.62	3.70	4.10	3.77	4.09	2.73	3.65
Q3	0.72	0.73	0.73	0.66	0.65	0.71	0.67	0.50	0.72	0.62	0.67	0.63	0.79	0.67
Med	0.05	0.04	0.05	-0.04	-0.05	-0.01	-0.02	-0.18	0.03	-0.06	-0.03	-0.03	0.12	-0.01
Q1	-0.82	-0.68	-0.65	-0.76	-0.73	-0.68	-0.70	-0.85	-0.65	-0.72	-0.67	-0.69	-0.56	-0.68
Min	-1.96	-3.09	-3.20	-4.13	-3.12	-2.89	-3.07	-2.23	-3.01	-3.50	-2.96	-3.41	-4.12	-3.18
Skew	0.22	0.04	0.02	0.01	0.12	0.10	0.18	0.30	0.07	0.25	0.21	0.23	-0.37	0.16

Q-Q Plots

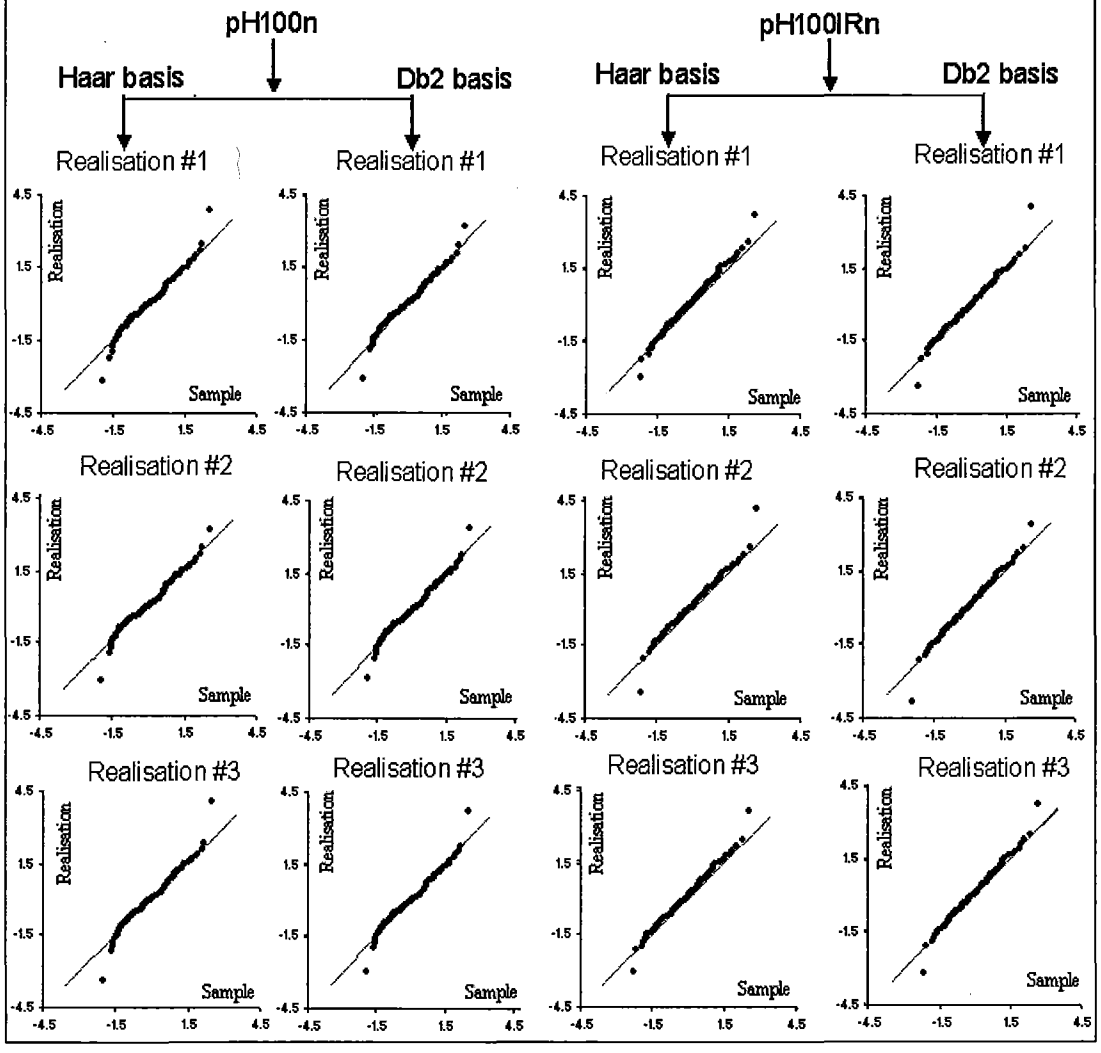


Figure 39: Summary statistics and Q-Q plots of three randomly selected realisations from *pH100n* and *pH100IRn* (simulated by DWSIM) against corresponding samples.

sample variance. The maxima/minima of all realisations are higher/lower than the corresponding sample maximum/minimum since this is a feature of Monte Carlo simulation and since the sample is only an approximate representative of the population. However, the skewness coefficients of all realisations are lower than the corresponding sample skewness coefficients.

7.2.3 Case Study 4: Applications of DWSIM to Anisotropic Samples

As in Case Study 2 in Chapter 7.1.3, the samples used to test the performance of DWSIM in the anisotropic case are *B64n* and *B64IRn*. Since the dimensions of the simulated realisations are required to be a power of 2, the simulations were carried out in a study region of size 64×64 then each realisation was trimmed to size 40×40. For each wavelet basis, three realisations were randomly selected from a set of 1000 realisations from each sample. The mosaic plots, semivariograms and the quantile-quantile plots of the realisations are shown in Figures 40, 41 and 42, respectively.

It can be seen from the mosaic plots in Figure 40 that the realisations simulated via the Haar basis have captured the features of both samples and the exhaustive data set *Berean*. Locations having high values at the top right corner of the study region can be seen in all of the realisations in columns 1 and 3 of Figure 40. The locations with low values associated with the banding in the direction $N55W$ at the middle left and near the bottom right corner of the samples *B64n* and *B64IRn* are reproduced. No artefacts are found in the mosaic maps of the realisations. In the case of the Db2 basis, locations associated with low and high values are also reproduced. However, the locations associated with the banding are better reproduced in the case of realisations from the ungridded sample *B64IRn* in column 4 than in the case of realisations from the gridded sample *B64n* in column 2. In both bases, the low extreme values of the

realisations are lower than the corresponding sample minimum and the high extreme values of the realisations are higher than the corresponding sample maximum.

Semivariogram models of both samples are reproduced in the Haar basis case. The experimental semivariograms for the normal scores of the realisations in columns 1 and 3 of Figure 41 are similar to the associated semivariogram models. The sill values in the directions of maximum continuity $N55W$ and minimum continuity $N35E$ are clearly distinguished. However, for the Db2 basis, only the semivariogram model of the ungridded sample *B64IRn* is approximately reproduced. The ranges in the direction of maximum spatial continuity $N55W$ of all semivariograms of realisations from *B64n* in column 2 are smaller than that of the associated model.

In terms of sample histogram reproduction, the summary statistics and the quantile-quantile plots in Figure 42 show that the means and variances of both samples are reproduced. Except for the variances of the realisations from *B64n* simulated using the Db2 basis, which are all lower than the sample variance, the variances of the other realisations fluctuate above and below the corresponding sample variance. For both wavelet bases, the minima (maxima) of all realisations are lower (higher) than the corresponding sample minimum (maximum). However, the medians and quartiles are close to the corresponding sample median and quartiles. The skewness coefficients of all realisations are close to the corresponding sample skewness coefficients, except for the case of Realisations #1 and #3 from *B64IRn* simulated using the Db2 basis. In general, in the anisotropic case, DWSIM reproduces sample histograms regardless of the wavelet basis and the sampling method.

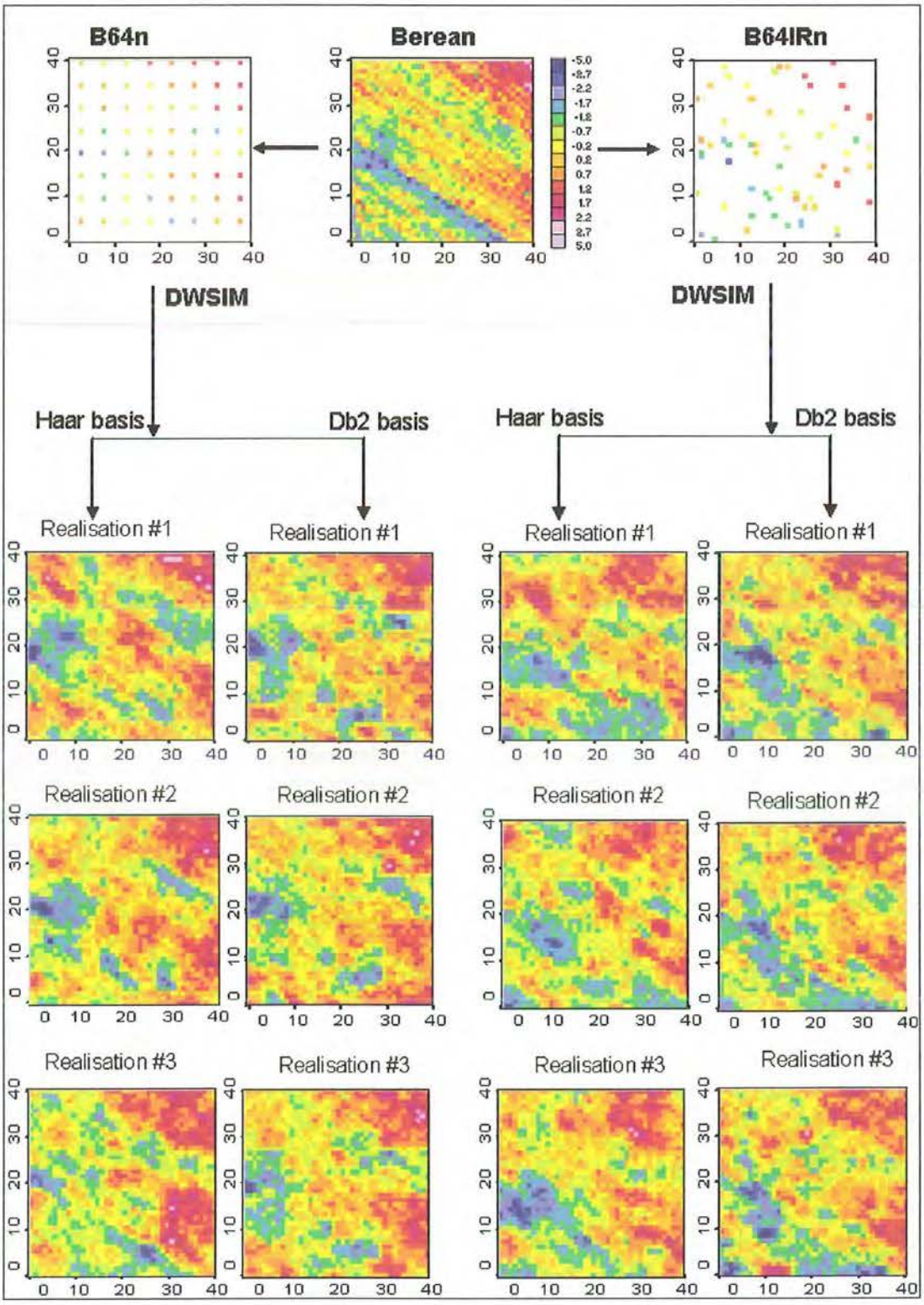


Figure 40: Mosaic plots of three randomly selected realisations from *B64n* and *B64IRn* simulated via DWSIM.

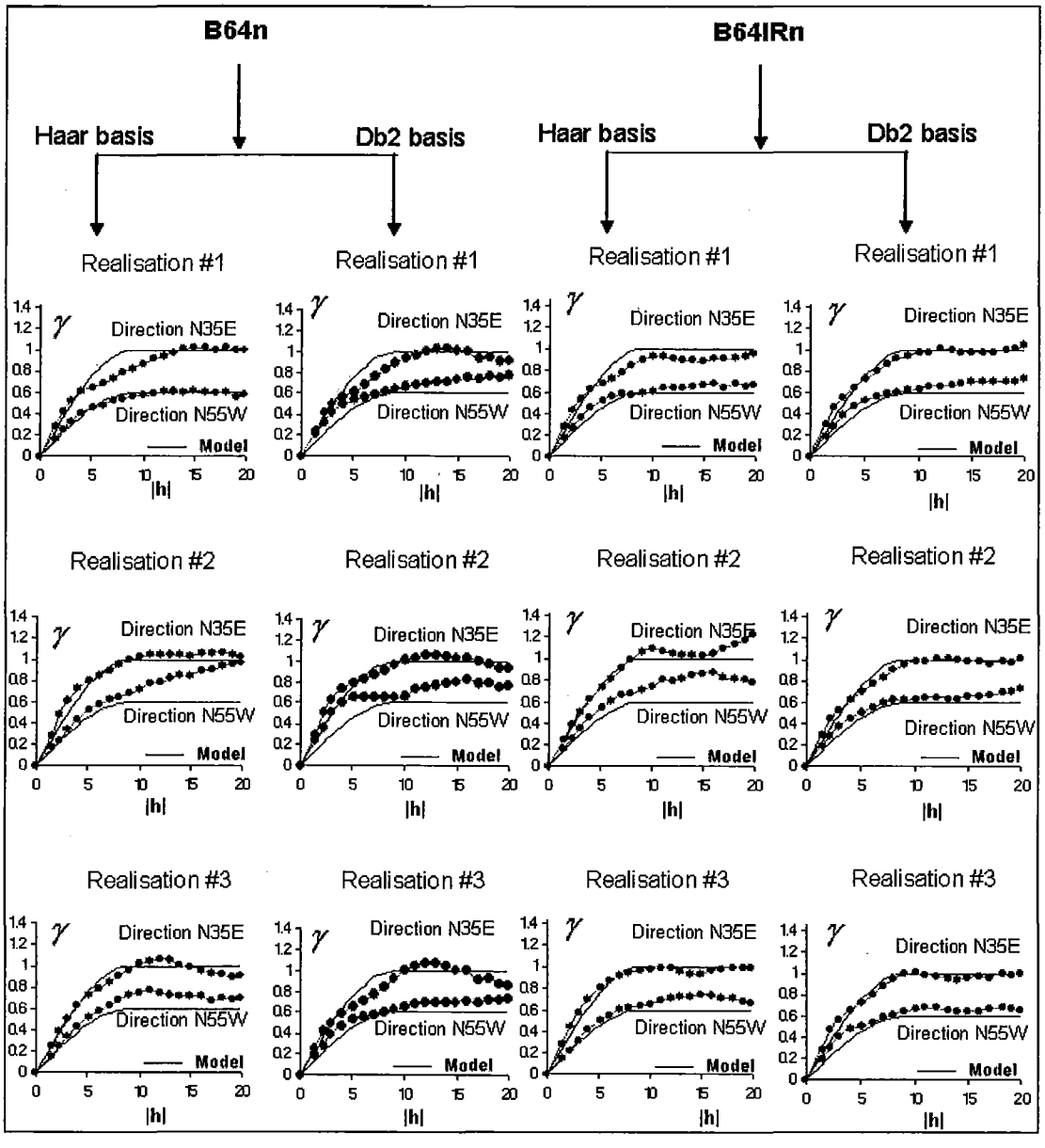


Figure 41: Semivariograms the three randomly selected realisations from *B64n* and *B64IRn* simulated via DWSIM.

Summary Statistics

	B64n							*B64IRn*						
	Sample	Realisations						Sample	Realisations					
		Haar			Db2				Haar			Db2		
Stats		1	2	3	1	2	3		1	2	3	1	2	3
Mean	0.19	0.10	-0.02	0.14	0.12	0.19	0.17	-0.05	-0.01	0.09	0.17	0.06	0.00	0.07
Var	1.03	1.03	1.02	1.09	0.94	0.92	0.92	1.02	0.85	1.02	0.98	0.97	0.93	1.06
Max	2.36	2.93	2.81	3.14	2.44	3.18	2.95	2.15	2.42	3.19	3.24	2.66	2.76	2.73
Q3	0.91	0.80	0.72	0.86	0.79	0.87	0.84	0.64	0.65	0.76	0.85	0.71	0.68	0.79
Med	0.12	0.10	-0.04	0.17	0.18	0.18	0.18	0.09	-0.03	0.13	0.19	0.08	0.01	0.12
Q1	-0.42	-0.59	-0.67	-0.58	-0.49	-0.41	-0.49	-0.87	-0.63	-0.52	-0.50	-0.53	-0.69	-0.56
Min	-2.42	-3.38	-3.61	-3.12	-3.40	-2.71	-3.39	-2.29	-2.88	-3.03	-3.04	-3.62	-2.68	-4.12
Skew	-0.13	-0.04	-0.21	-0.15	-0.15	-0.20	-0.10	-0.09	-0.03	-0.23	-0.07	-0.28	-0.04	-0.37

Q-Q Plots

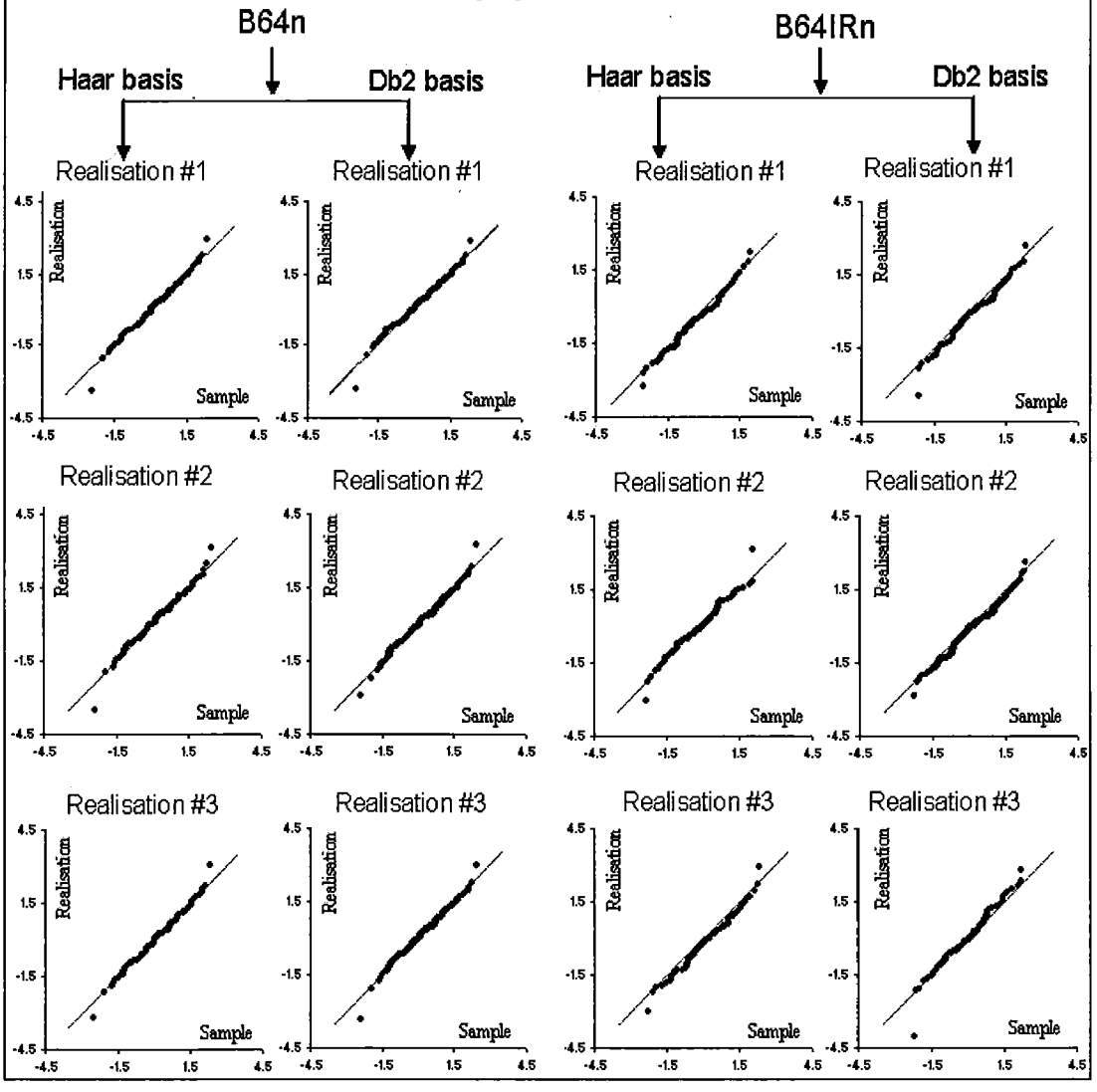


Figure 42: Summary statistics and Q-Q plots of three randomly selected realisations from *B64n* and *B64IRn* (simulated via DWSIM) against corresponding samples.

7.2.4 General Comments on DWSIM

We have seen in case studies 3 and 4 that DWSIM using the Haar basis reproduces the semivariogram model and sample histogram in both isotropic and anisotropic cases regardless of the sampling method. Even though DWSIM using the Db2 basis reproduces the histograms of all samples, it fails to appropriately reproduce the spatial continuity of the anisotropic gridded sample *B64n*. In the next Chapter, we will investigate the performance of DWSIM in more detail by quantitatively comparing the global and local accuracy of the simulated realisations obtained via DWSIM with those of realisations simulated via PWSIM. The computational effort for these algorithms is also discussed to compare the computational efficiency of two algorithms. The results of the comparison will be used to determine the appropriate wavelet basis for DWSIM.

8 COMPARISON OF PWSIM AND DWSIM

DWSIM and PWSIM are similar in that they are both multi-level wavelet-based simulation algorithms and both exploit the second-order stationarity property of the underlying random function. Both algorithms start the simulation from an initial scaling image at some coarse scale and then reconstruct the scaling image at the finer scale using the correlation of scaling/wavelet coefficients.

However, the algorithms differ in the methods used to obtain the conditional simulated realisations. Firstly, the initial scaling image in DWSIM is estimated from the conditioning data via the discrete wavelet transform while in PWSIM the initial scaling image is simulated using the Choleski decomposition. Secondly, in DWSIM only the scaling coefficients at the coarser scale are used to simulate the scaling coefficients at the finer scale while in PWSIM the scaling coefficients at the finer scale are computed from the simulated scaling and wavelet coefficients at the coarser scale using the inverse discrete wavelet transform. The method used to estimate the initial scaling image and to simulate the scaling coefficients at the finer scale in DWSIM makes it possible to condition the data directly.

In Chapter 7, we illustrated briefly the performance of PWSIM and DWSIM. In this Chapter we will quantitatively assess the performance of the two algorithms for each wavelet basis. The performance measures used include the accuracy and goodness coefficients A and G, the average variance $\overline{V}$ of the local distributions, and MAD and MAE values described in Chapter 4. The realisations used to compute these measures are those from *pH100n* and *pH100IRn* in the isotropic case and those from *B64n* and *B64IRn* the anisotropic case.

8.1 Accuracy, Goodness and Precision

The *accuracy*, *goodness* and *precision* of the simulation are visually assessed via the accuracy plots and quantitatively assessed via the accuracy coefficient A and the goodness coefficient G in conjunction with the average variance $\overline{V}$.

Using formula (122), for each sample the computation of the accuracy, goodness and precision coefficients are based on the local conditional cumulative probability distribution functions modelled from 1000 realisations with the jackknife data from Chapter 5. The accuracy plots for each sample, together with the coefficients, are shown in Figure 43 with the plots in the case of PWSIM in rows 1 and 2 and those in the case of DWSIM in rows 3 and 4.

In the isotropic case, since the accuracy coefficients A and goodness coefficients G computed from realisations simulated via PWSIM in row 1, column 1 and row 2, column 2 are higher than those computed from realisations simulated via DWSIM in row 3, column 1 and row 4, column 2, PWSIM performs better than DWSIM for both wavelet bases. For the Haar basis, in both samples, PWSIM performance is only slightly better than DWSIM performance because the accuracy and goodness coefficients in the case of PWSIM are only slightly higher than those in the case of DWSIM. For the Db2 basis, PWSIM performance is much better than DWSIM performance because the accuracy and goodness coefficients computed in the case of PWSIM are much higher than those computed in the case of DWSIM.

The setback for the higher accuracy and goodness coefficients in the case of PWSIM is that the average variances $\overline{V}$ computed from realisations simulated via PWSIM are higher than those computed from realisations simulated via DWSIM. Similar to the case of the accuracy and goodness coefficients, the differences between the average variances computed from realisations simulated via the two algorithms

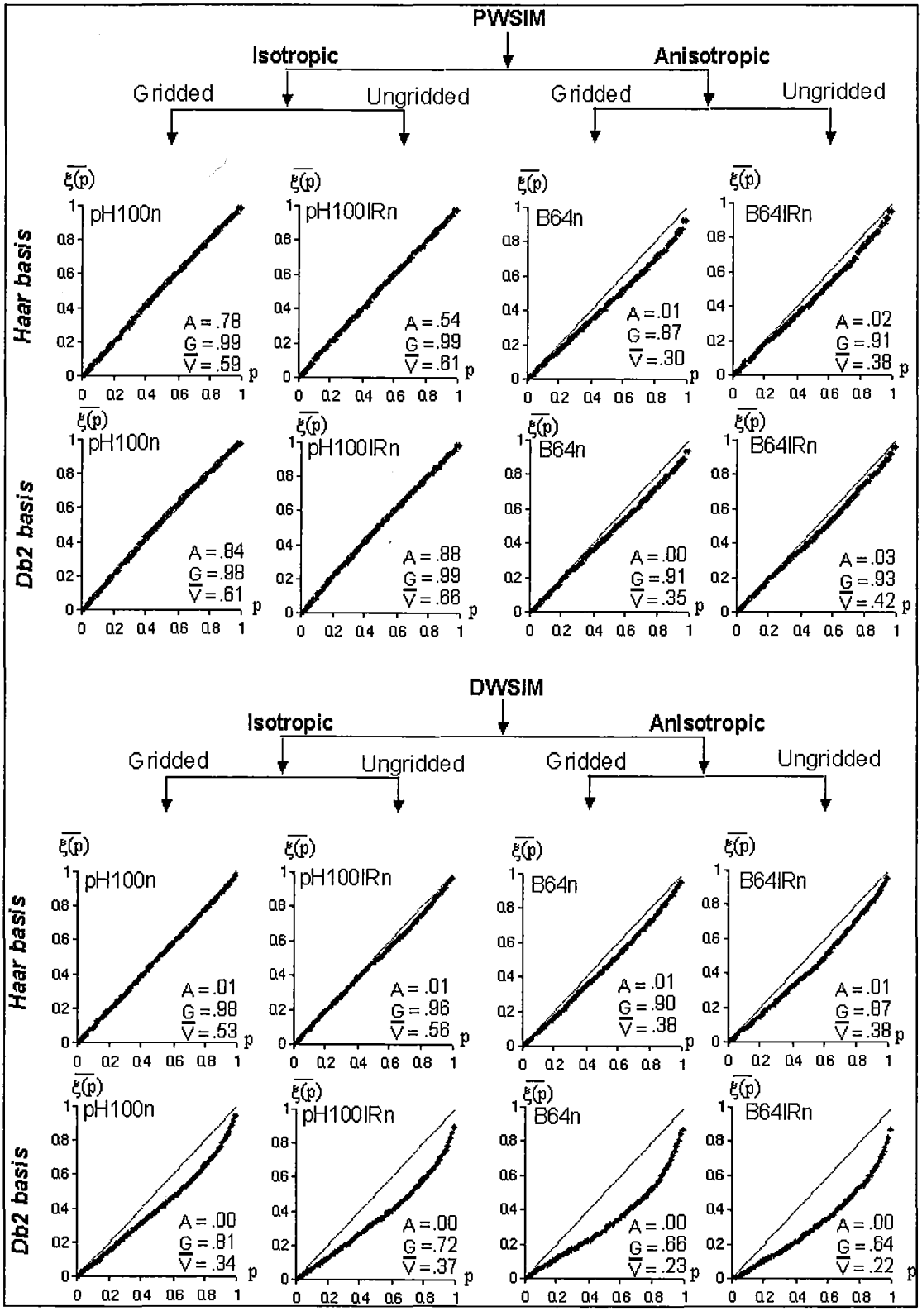


Figure 43: Accuracy plots obtained using PWSIM and DWSIM.

are negligible for the Haar basis and considerable for the Db2 basis. This indicates that the results of the simulation using PWSIM are less precise than those in the case of DWSIM.

In the anisotropic case, the accuracy plots in columns 3 and 4 of Figure 43 indicate that the performance of PWSIM and DWSIM are approximately equivalent if the Haar basis is used. Since the accuracy coefficients A are close to 0, we focus only on the goodness coefficient G and the average variance $\overline{V}$. For the gridded sample *B64n*, the goodness coefficient G, in row 1, computed from realisations simulated via PWSIM are lower than that computed from realisations simulated via DWSIM in row 3. The opposite is true for the average variance. For the ungridded sample *B64IRn*, the coefficient A is higher in the case of PWSIM while the coefficient $\overline{V}$ is the same.

Similar to the isotropic case, the performance of PWSIM is much better than the performance of DWSIM if the Db2 basis is used. The coefficients G in both samples in the case of DWSIM in row 4 are much lower than those in row 2 in the case of PWSIM. The average variances $\overline{V}$ in the case of DWSIM are also much lower than those in the case of PWSIM.

In comparing the impact of wavelet basis with regard to the same algorithm, PWSIM using the Db2 basis performs slightly better than PWSIM using the Haar basis since the coefficients A and G in the case of the Db2 basis are slightly higher than those in the case of the Haar basis. The increase of these coefficients is traded by the increase of the average variances. On the other hand, DWSIM using the Haar basis performs much better than DWSIM using the Db2 basis.

The precision of DWSIM and PWSIM with regard to the two wavelet bases is visualised via the plots of the average width $\overline{W}$ of the probability intervals that contain the true values against the probability p in Figure 44. It can be seen that for both

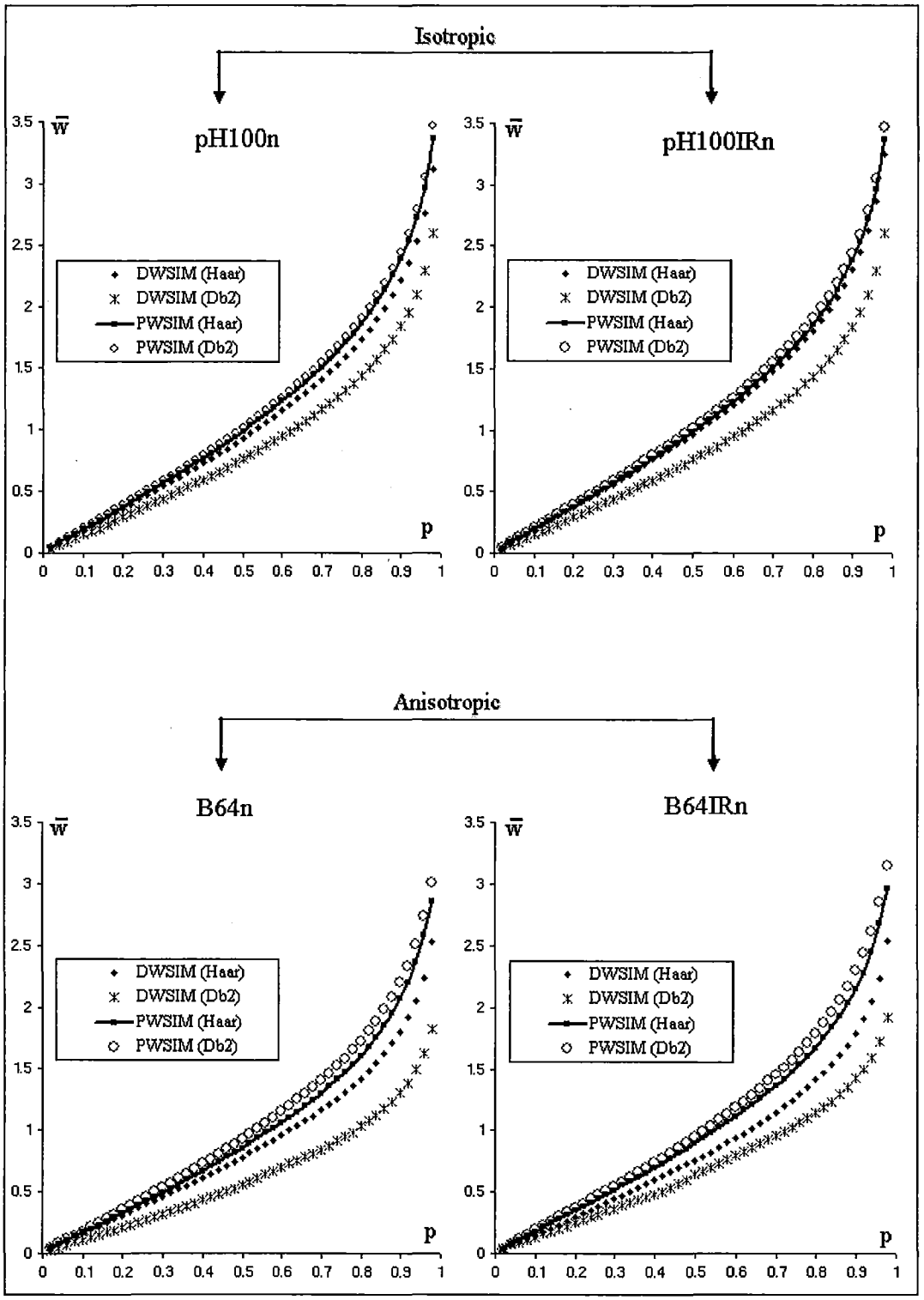


Figure 44: Plots of average probablity intervals of realisations simulated via DWSIM and PWSIM.

isotropic and anisotropic cases the widths of the probability intervals modelled by realisations simulated via PWSIM are wider than those in the case of DWSIM. Since the wider the probability widths the more true values they can contain, the accuracy coefficients in the case of PWSIM are higher than those in the case of DWSIM. Similarly, since the widths of probability interval modelled by realisations simulated via PWSIM using the Db2 basis are larger than those modelled by realisations simulated using the Haar basis, the accuracy coefficients obtained via PWSIM using the Db2 basis are higher. The opposite is true for DWSIM.

8.2 Sample Histogram and Semivariogram Model Reproduction

The reproduction of the histogram and semivariogram model of each sample will be assessed via the MAD and MAE values of 1000 conditional realisations. The MAD is the mean of the absolute deviations between the quantiles of a simulated realisation and the quantiles of the associated sample. The MAE is the mean of the relative error between the experimental semivariogram of a simulated realisation and the associated semivariogram model. Summary statistics for the MAD values, computed using 20 quantiles, and the MAE values, computed from 18 lags in the isotropic case and 9 lags in the anisotropic case using lag spacing of 1 measurement unit, are displayed by means of the boxplots in Figures 45 and 46 respectively. The boxplots for the MAD and the MAE of realisations simulated using the Db2 basis are on the right of the Figures with those for the MAD and MAE values computed from realisations simulated via PWSIM being shaded.

Since the medians of the MAD values shown by the boxplots in Figure 45 are close to 0 for all cases, for each algorithm the histogram of each sample is reproduced

regardless of the wavelet basis used. By comparing the boxplots in two algorithms, one can see that except for the median of the MAD values computed for realisations from *pH100IRn* simulated using the Haar basis, the medians of the MAD values in the case of DWSIM are lower than those in the case of PWSIM. Therefore, in general, in terms of sample histogram reproduction DWSIM performs better than PWSIM.

With regard to the wavelet basis used, the sample histogram is better reproduced in the case of the Db2 basis than in the case of the Haar basis for DWSIM. All the medians on the right of Figure 45 are smaller than those on the left. For PWSIM, on average the use of the wavelet basis does not have a clear impact on sample histogram reproduction. In the isotropic case, the histogram of the gridded sample is better reproduced in the case of the Db2 basis than in the case of the Haar basis. The opposite is true for the histogram of the ungridded sample. In the anisotropic case, the use of the Db2 basis worsens sample histogram reproduction regardless of the sampling method.

In terms of semivariogram model reproduction, in the isotropic case, for both wavelet bases DWSIM performs better than PWSIM for the gridded sample *pH100n*, and the opposite is true for the ungridded sample *pH100IRn*. In the anisotropic case, PWSIM performs better than DWSIM in the anisotropic case regardless of the wavelet basis and sampling method. In fact, for DWSIM using the Db2 basis, the semivariogram model of anisotropic sample *B64n* is not appropriately reproduced. It can be seen on the right of Figure 46 that the minimum and median of the MAE values computed for realisations simulated via DWSIM (from sample *B64n*) using the Db2 basis is considerately higher than 0.0.

With regard to the impact of the wavelet basis on each algorithm, in the isotropic case the use of the Db2 basis slightly improves the performance of both algorithms

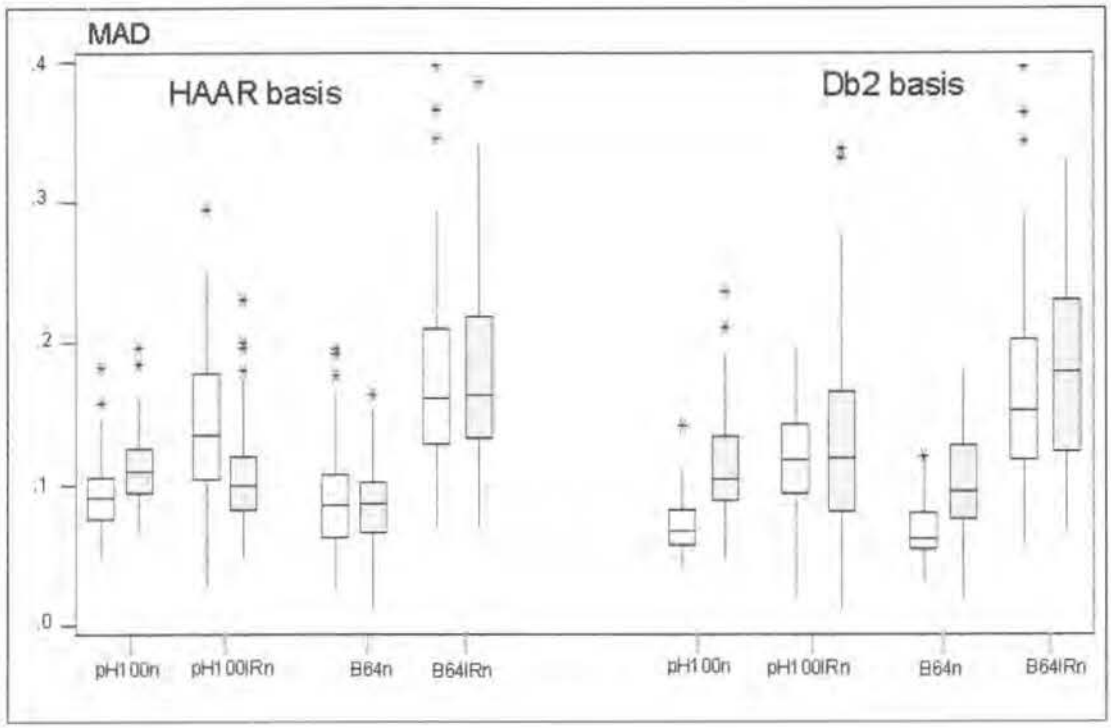


Figure 45: Boxplots for MAD values computed from realisations simulated via PWSIM (shaded) and DWSIM.

in the case of the gridded sample *pH100n*. The opposite is true for the ungridded sample *pH100IRn*. In the anisotropic case, PWSIM using the Db2 basis performs better than PWSIM using the Haar basis, regardless of the sampling method. The opposite applies for DWSIM.

We have shown from the summary statistics for the MAE values that PWSIM, using either the Haar or the Db2 basis, and DWSIM using the Haar basis reproduce the spatial continuity of the corresponding sample regardless of the sampling method and the spatial structure. DWSIM using the Db2 basis only appropriately reproduces the semivariogram models in the isotropic case. We will now visualise the reproduction of spatial continuity by plotting the experimental semivariograms for the realisations simulated via the two algorithms, together with the semivariogram models.

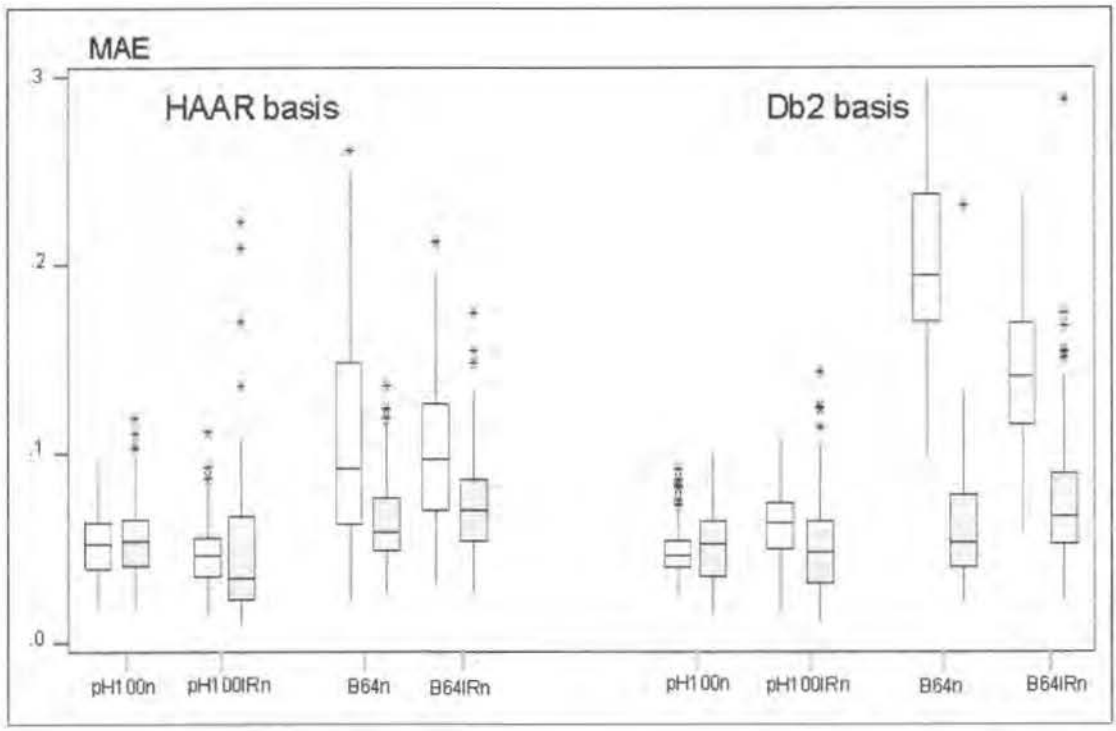


Figure 46: Boxplots for MAE values computed from realisations simulated via PWSIM (shaded) and DWSIM.

For readability, only the experimental semivariograms of the first 200 realisations from each sample are plotted. The experimental semivariograms and the associated semivariogram models in the isotropic case are shown in Figure 47 and those in the anisotropic case in Figure 48. The plots in Figure 47 indicate that, in both algorithms the semivariograms for the realisations from *pH100n* and *pH100IRn* are similar to the corresponding semivariogram models. They fan out about the models with those associated with the maximum errors lying far above or below the models and those with minimum errors overlapping the models. In the case of PWSIM the range of the fluctuation is similar for both bases. In the case of DWSIM, the variability of the semivariograms for realisations simulated via the Db2 basis is much lower than that in the case of the Haar basis.

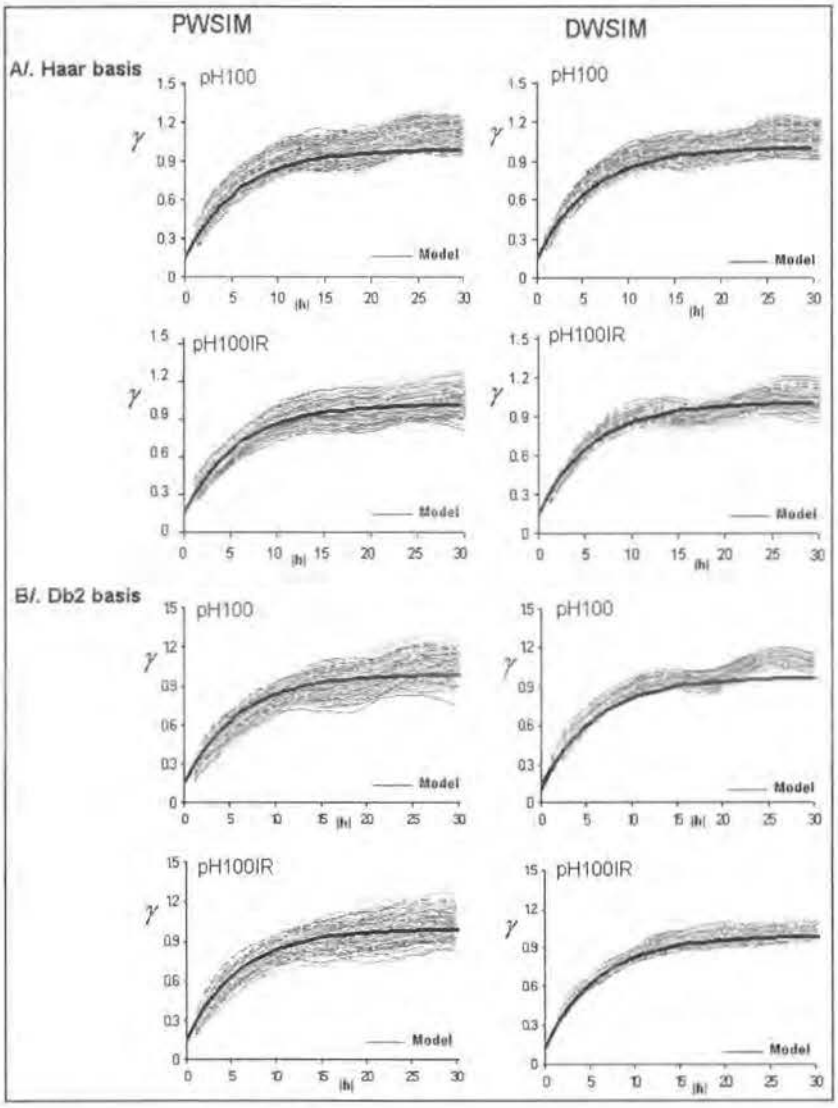


Figure 47: Experimental semivariograms for realisations from isotropic samples simulated via PWSIM and DWSIM.

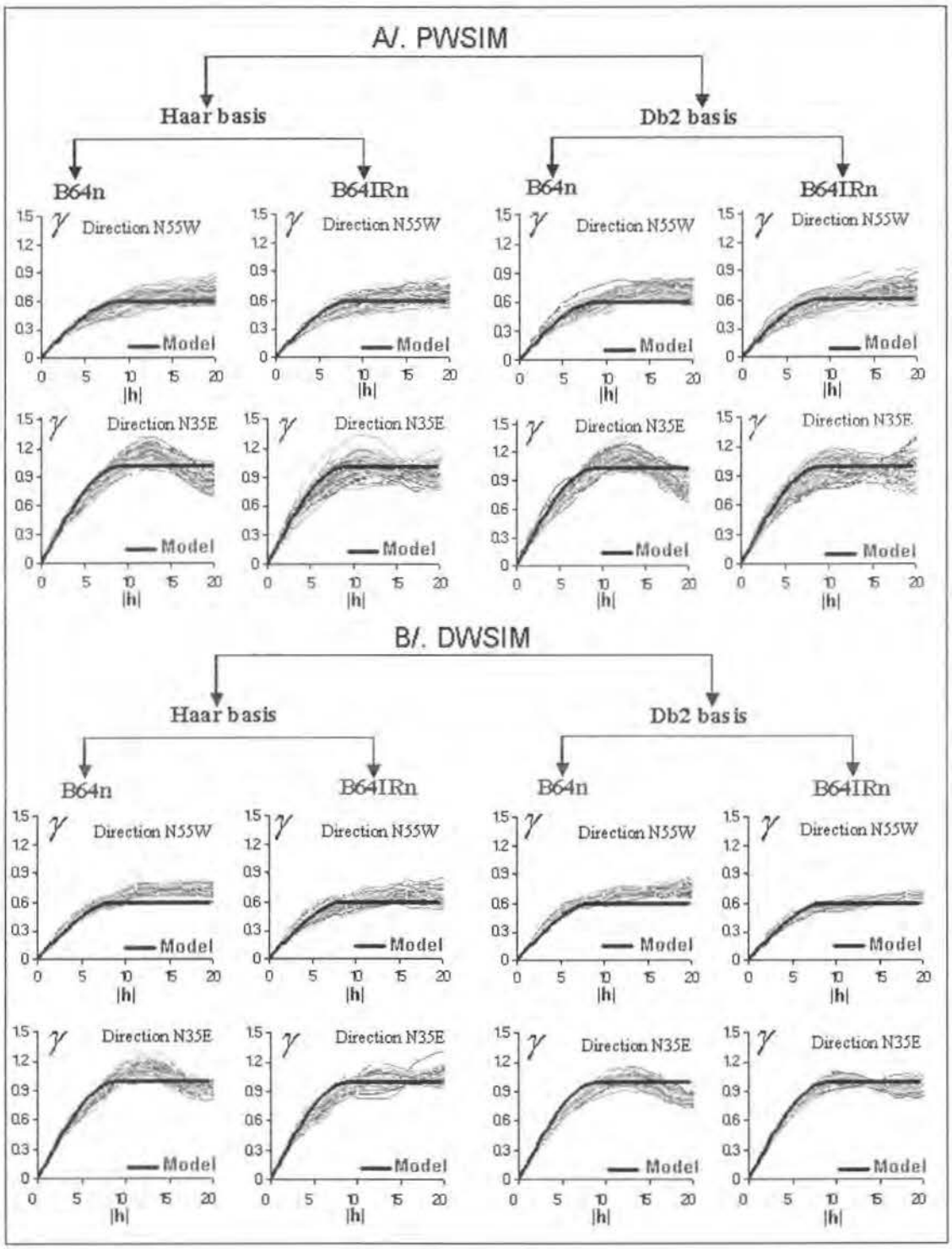


Figure 48: Experimental semivariograms for realisations from anisotropic samples simulated via PWSIM and DWSIM.

In the anisotropic case, the plots in Figure 48 show that, for both wavelet bases, the spatial continuity is better reproduced for realisations simulated via PWSIM than for those simulated via DWSIM. The semivariogram models of *B64n* and *B64IRn* are covered completely by the experimental semivariograms of the realisations simulated via PWSIM. For DWSIM, the spatial continuity is better reproduced in the case of sample *B64IRn* than in the case of *B64n*. Regardless of the wavelet basis, the sill values of the experimental semivariograms of the realisations from *B64n* in the direction of maximum continuity *N55W* are higher than the sill value of the model (in that direction). One reason for the better performance of PWSIM compared with DWSIM in terms of spatial continuity reproduction is that realisations generated using PWSIM have larger local variances. This leads to the larger fluctuation of the semivariograms of the realisations; hence, these semivariograms better cover the associated semivariogram model. The plots in Figure 48 also show that the variability among the experimental semivariograms for realisations simulated via PWSIM is similar for both wavelet bases. For DWSIM, the variability among the experimental semivariograms of realisations simulated via DWSIM using the Haar basis is higher than the variability among those of realisations simulated via DWSIM using the Db2 basis.

In summary, if the Haar basis is used, in terms of accuracy and goodness coefficients, PWSIM performs better than DWSIM in the isotropic case and DWSIM performs as well as PWSIM in the anisotropic case. The drawback for the higher accuracy and goodness coefficients in the case of PWSIM compared with the case of DWSIM is the bigger average variance. Since the average variance is bigger, the spread of the local probability distributions is larger; therefore, the proportion of the true values falling within the probability intervals becomes higher. In terms of

histogram and semivariogram reproduction, the performances of the two algorithms are equivalent in the isotropic case. In the anisotropic case, DWSIM performs better than PWSIM if the MAD values are compared and the opposite is true if the MAE values are compared.

If the Db2 basis is used, PWSIM performs better than DWSIM for both isotropic and anisotropic cases if the accuracy and goodness coefficients are compared and the opposite is true if the MAD values are compared. In terms of semivariogram model reproduction, the DWSIM performance is equivalent to PWSIM performance in the isotropic case and PWSIM performance is better than DWSIM performance in the anisotropic case.

The performance comparison between PWSIM and DWSIM, with regard to the same wavelet basis and spatial structure, that is, PWSIM using the Haar (Db2) basis compared with DWSIM using the Haar (Db2) basis and PWSIM applied to isotropic (anisotropic) sample compared with DWSIM applied to isotropic (anisotropic) sample, is shown in Table 5. The + in the table indicate the more efficient algorithm whereas the = indicate an equivalent performance.

	PWSIM				DWSIM			
	Haar		Db2		Haar		Db2	
	Iso.	Ani.	Iso.	Ani.	Iso.	Ani.	Iso.	Ani.
Accuracy/Goodness	+	=	+	+		=		
Average variance		=			+	=	+	+
MAD	=				=	+	+	+
MAE	=	+	=	+	=		=	

Table 5: Performance comparison between PWSIM and DWSIM.

With regard to the impact of wavelet basis used in each algorithm, PWSIM using the Db2 basis performs better than PWSIM using the Haar basis if the performance measure is based on the accuracy and goodness coefficients. There is no clear trend for other measures. For DWSIM, except for the performance measure based on the MAD values, DWSIM using the Haar basis performs better than DWSIM using the Db2 basis.

8.3 The Impact of Trimming the Simulation Grid on Histogram Reproduction

In this section we investigate the impact of trimming the simulation grid on histogram reproduction. Since both DWSIM and PWSIM require that the dimensions of the study region be a power of 2, the simulation is usually carried out in a region that is larger than the actual one. At the conclusion of the simulation, both DWSIM and PWSIM report only the simulated values within the study region. As data are only available within the study region, one can expect that trimming the realisations improves the performance of the algorithms based on the reproduction of the corresponding sample histogram.

The left and middle columns of Figure 49 show the mosaic plots of the realisations from *B64IRn* after and before trimming the simulation grid, respectively. The realisation obtained via DWSIM is shown in the second row whereas the realisation obtained via PWSIM is shown in the third row. The study region in this case is a 40×40 grid whereas the simulation has to be carried out in a 64×64 grid. The boxplots in column 3 of Figure 49 show the summary statistics of the MAD values computed from 1000 realisations simulated using DWSIM (in the second row) and PWSIM (in the third row) with the shaded boxplots indicating summary statistics

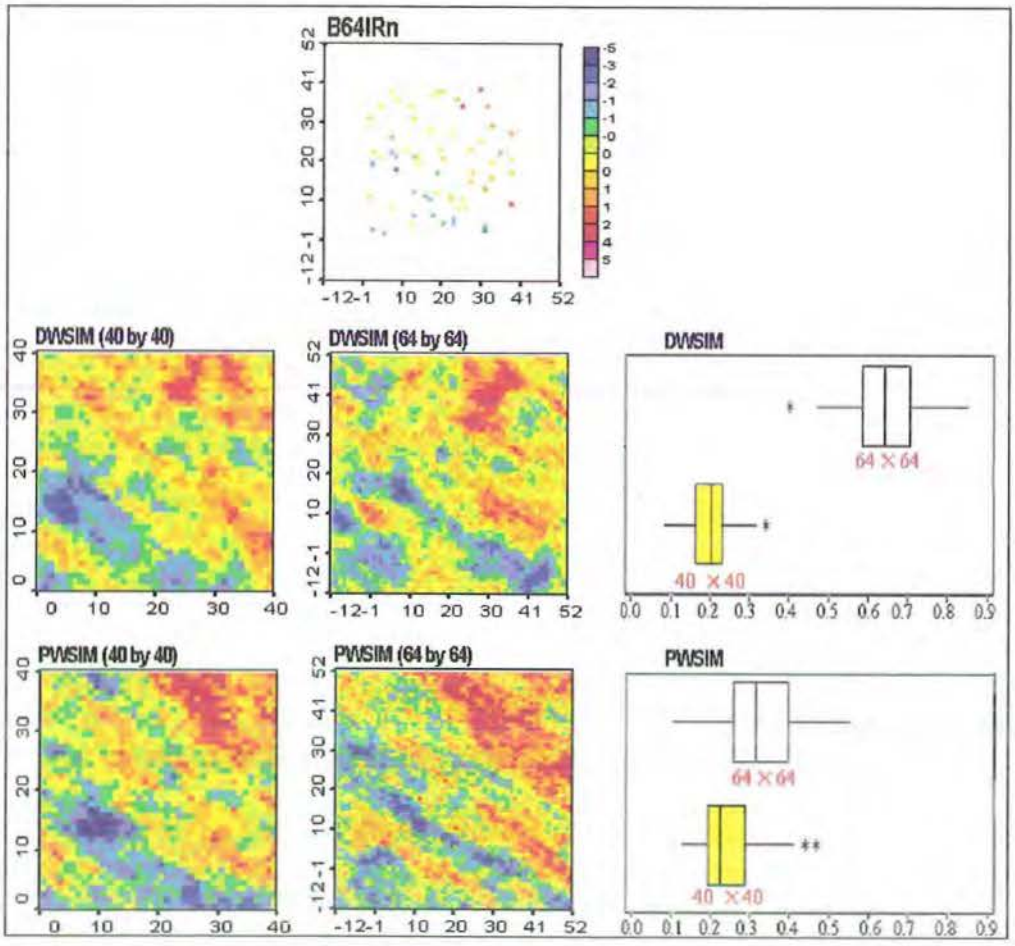


Figure 49: Mosaic maps and boxplots illustrating the impact of trimming simulation grid on histogram reproduction.

for MAD values after the simulation grid is trimmed to the grid size of 40×40. It is clear that the sample histogram is better reproduced after trimming. In the case of PWSIM, the MAD values before trimming are only slightly larger than the MAD values after the trimming. However, in the case of DWSIM the MAD values before trimming are significantly larger than those after trimming. The reason for the difference in the impact of trimming the simulation grid on the two algorithms comes from the method used to condition the data. In PWSIM, a larger search radius and number of conditioning data have to be used in the kriging step to allow the grid

nodes outside the study region to be simulated in terms of sufficiently many conditioning values within the study region. The cost for this is a higher computational effort. In DWSIM, due to the size of the filter length and the conditioning method, the simulated values at the nodes outside the study region usually depend only on a very small number of conditioning values close to the edges. Therefore, if these nodes are included in the realisations, the deviation of the realisation histogram and the sample histogram is larger.

8.4 The Impact of the Size of the Initial Scaling Image in PWSIM

In DWSIM, the reconstruction of the original scaling image is started at the scaling image all of whose grid nodes are estimated. PWSIM, on the contrary, can start the reconstruction at any coarse scale as long as the covariance matrix for the scaling coefficients is small enough to make the Choleski decomposition feasible. Different starting scales result in different sizes of the initial scaling image. In the comparison we have carried out previously, as discussed in Chapter 7, the simulation is started at an scaling image of size 4×4 but there is no reason that PWSIM cannot begin with an initial scaling image of smaller or of larger size. Now we investigate the impact of the size of the scaling image on the performance of PWSIM in the isotropic case.

Since the size of the exhaustive data set *pHsoil* is 61×61, the size of original scaling image ($j = 0$) for the simulation is extended to 64×64. At the end of the simulation, the program trims off the simulated realisation back to size 61×61. Since the size of the original scaling image for the simulation is 64×64, the maximum number of times the discrete wavelet transform can be applied to this image is $log_2(64) = 6$. In other words, the coarsest scale at which PWSIM can start is scale 2^6 when the size of

the initial scaling image is 1×1. In this case, the initial scaling image is too small to carry enough the information of the original scaling image. Therefore, the simulation has to start at some finer scale so that the size of the covariance matrix for the scaling coefficients is small enough so that the decomposition can be feasibly applied and the size of the scaling image is large enough to contain the correlation from the original scale.

As discussed, the results we have obtained in this Case Study so far are based on the initial scaling image of size 4 by 4, that is the simulation starts at scale 2^4. The results may be worse or better if the simulation starts at some different scale. Therefore, we will investigate the performance of PWSIM using the Haar basis when the simulation starts at one scale coarser 2^5 (the size of the initial scaling image is 2×2) and at one scale finer 2^3 (the size of the initial scaling is 8×8). In each case 1000 realisations from each sample were generated. The mosaic plots of three randomly selected realisations from each sample in the case when the starting scale is 2^5 are shown in Figure 50B while those for when the starting scale is 2^3 are shown in Figure 50C. The maps in Figures 50B and 50C indicate that the realisations have captured the features of the exhaustive data set and of the associated samples regardless of the starting scale. The regions of high and low values seen from the mosaic plots of the exhaustive data set *pHsoiln* and of the samples *pH100n* and *pH100IRn* in Figure 50A are reproduced in these mosaic plots. No indication of artifacts is apparent in the plot of any realisation.

We have investigated visually three typical realisations simulated via PWSIM corresponding to different starting scales. We now explore the impact of the size of the initial scaling image on the performance of the algorithm using quantitative measures. These measures include the MAD and MAE values and the coefficients for

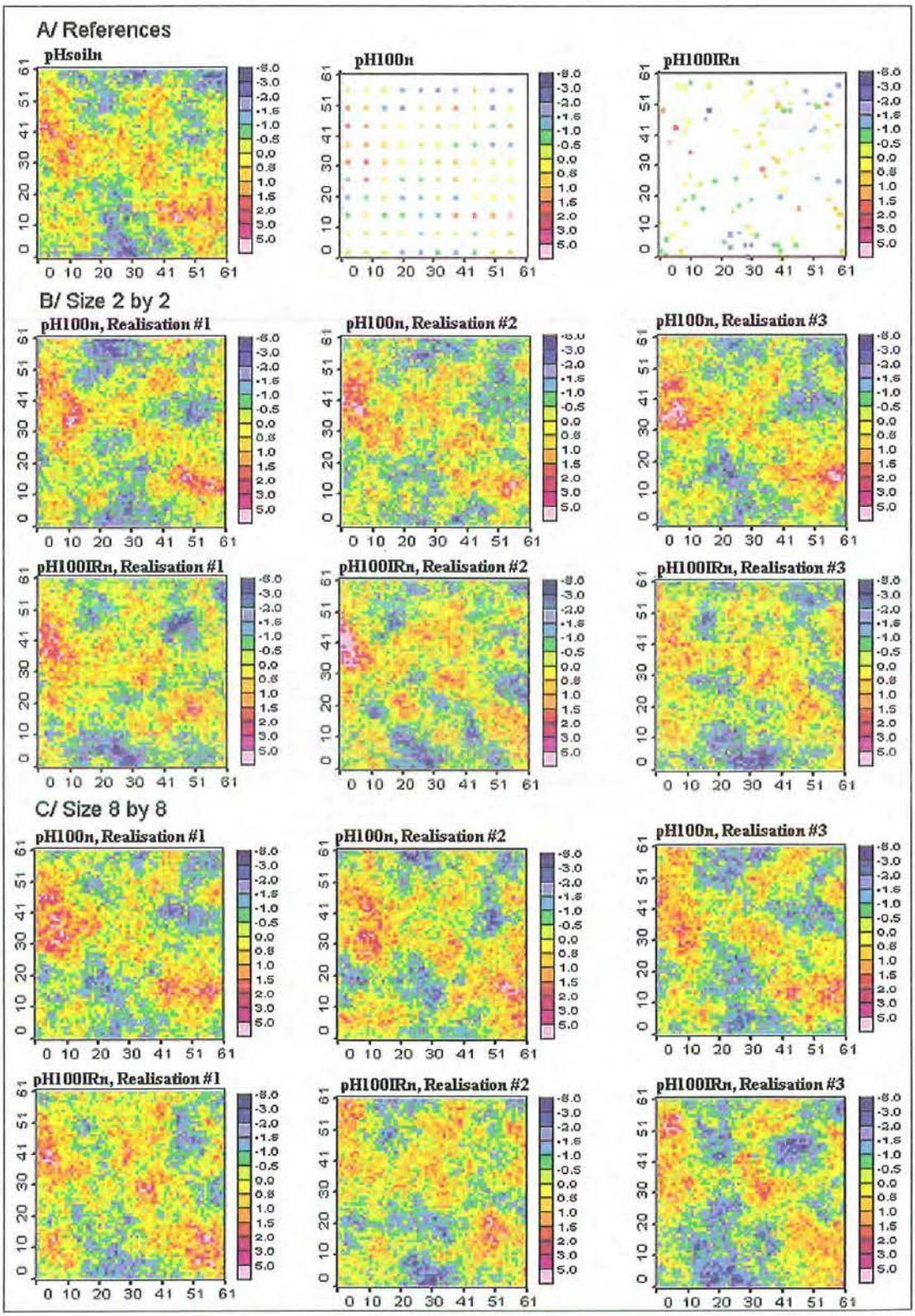


Figure 50: Mosaic plots of randomly selected realisations from *pH100n* and *pH100IRn* associated with initial scaling image of size 2×2 and 8×8.

evaluating the accuracy and goodness of the simulation.

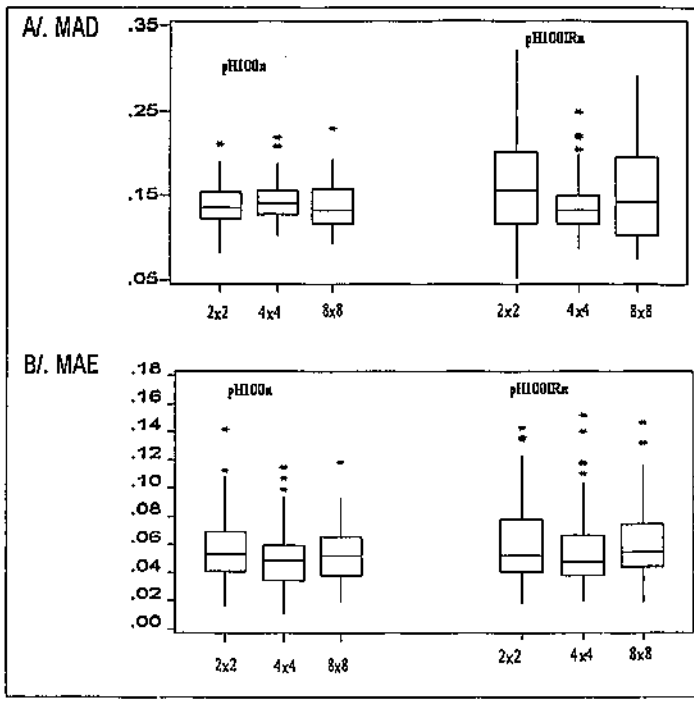


Figure 51: Boxplots for MAD and MAE values associated with different initial scaling image size.

It can be seen from Figure 51 that the size of the initial scaling image has little impact on the reproduction of the sample histograms and semivariogram models. For sample histogram reproduction, the summary statistics for MAD values (computed using 20 quantiles) of 1000 realisations of each sample shown in Figure 51A show that these values vary with the starting scale. However, there is no trend of improvement;

the reproduction of the sample histogram does not improve in accordance with the size of the initial scaling image. In fact, only the range of the MAD values changes. The medians of MAD values associated with the initial scaling image of size $2 \times 2, 4 \times 4$ and 8×8 in the case of sample *pH100n* are approximately equal to 0.130, 0.140 and 0.125; and those in the case of sample *pH100IRn* are 0.150, 0.130 and 0.140, respectively. The differences among these medians are small. Similarly, regarding semivariogram model reproduction, the summary statistics for 1000 MAE values (computed from 18 lags using a lag spacing of 1 unit) of both samples in Figure 51B indicate that increasing the size of the initial scaling image has little impact on the MAE values.

The accuracy plots in Figure 52 show that the size of the initial scaling image has only a small impact on the accuracy, goodness and precision of the simulation. The accuracy plots obtained from realisations of *pH100n* in Figure 52A and of *pH100IRn* in Figure 52B show that, in general, increasing the size of the initial scaling image slightly reduces the spread of the local distributions, in other words reduces the uncertainty of the results. There is no trend for the accuracy or the goodness of the distributions. As can be seen from the accuracy plots for realisations of *pH100IRn*, the average variance $\overline{V}$ slightly decreases when the size of the initial scaling image increases. However, this change is not very great. For sample *pH100n*, the coefficient $\overline{V}$ equals 0.54 for the initial scaling image of size 2×2, then increases to 0.59 for the initial scaling image of size 4×4 and reduces to 0.53 for the initial scaling image of size 8×8. For sample *pH100IRn*, the coefficient $\overline{V}$ decreases from 0.62 for the initial scaling image of size 2×2, then 0.61 for the initial scaling image of size 4×4 and 0.60 for the initial scaling image of size 8×8. If we disregard the middle graph in the first row of Figure 52A, the precision of the simulation increases with the size of the initial scaling image.

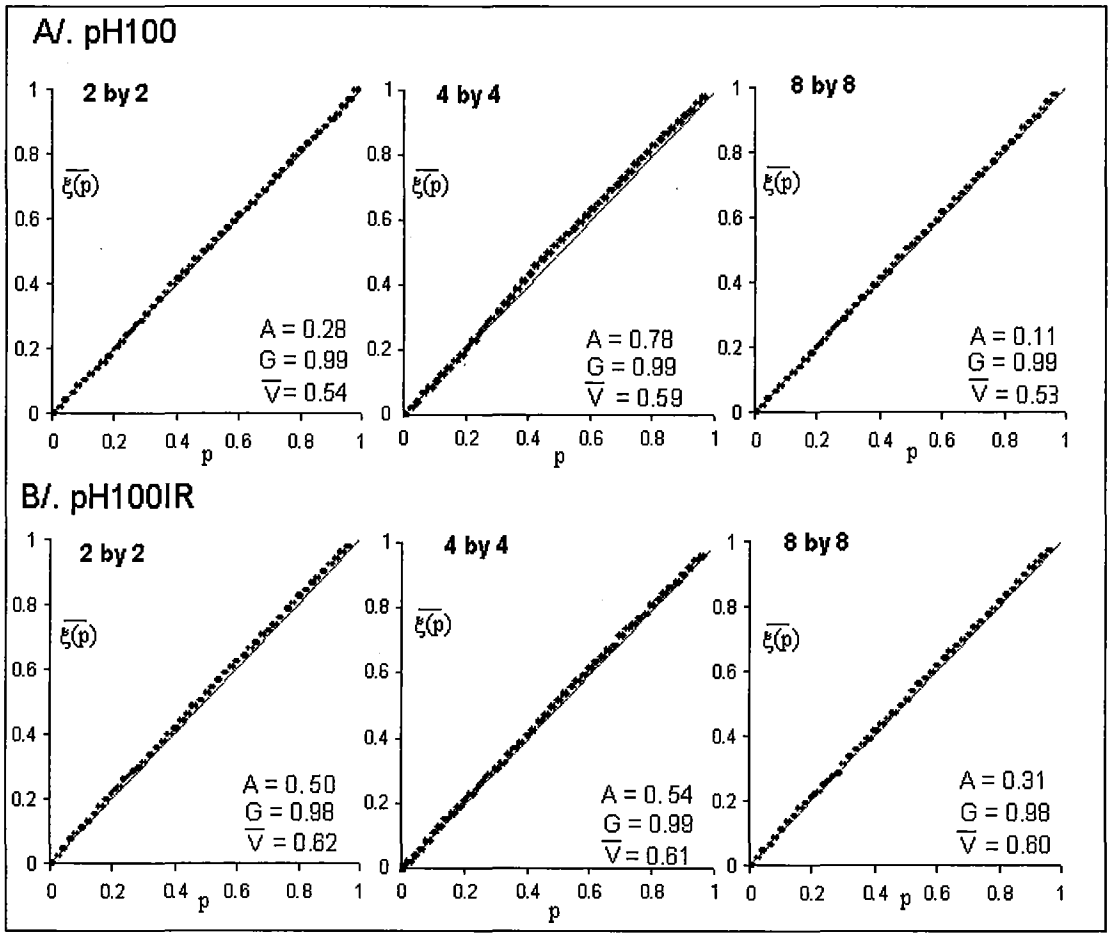


Figure 52: Accuracy plots for realisations of PWSIM with different initial scaling image sizes.

In summary, since the starting level of PWSIM has little impact on the performance of PWSIM, it can be ignored.

8.5 Computational Effort for PWSIM and DWSIM

In this Section we compute the computational effort needed by PWSIM and DWSIM in the simulation of one realisation. As an illustration we count the number of algebraic operations used to simulate one realisation of the sample *pH100n*. From a debug file created by the computer program, the size of initial scaling image in the case of DWSIM for this example is 8×8, therefore, for consistency we assume that PWSIM also starts the simulation with the initial scaling image of size 8×8 that is associated with $j = 3$.

8.5.1 Computational Effort for PWSIM

Let M be the number of vanishing moments of the wavelet basis used, the computational effort used in PWSIM includes

1. The computation of the covariance tables requires $(5603 - 460M + 12M^2) \times (80M^4 - 1) + 66564$ operations.

2. The simulation of the initial scaling image requires 93472 algebraic operations.

3. The computation of the weights requires $3 \times (\frac{1}{3}(2M+1)^6 + \frac{25}{2}(2M+1)^4 - \frac{5}{6}(2M+1)^2 + 3)$ operations.

4. The simulation of the wavelet coefficients requires $8064\,(2M+1)^2 + 4032$ operations.

5. The inverse discrete wavelet transform requires $64512M^2 - 5376$ additons and multiplications.

6. Conditioning the data: assume that 16 conditioning data are used to compute the kriging estimate at one unknown location this step requires 1.154×10^7 additions and multiplications.

Step	N° of operations
1	$(5603 - 460M + 12M^2)(80M^4 - 1) + 66564$
2	93472
3	$3 \times (\frac{1}{3}(2M+1)^6 + \frac{25}{2}(2M+1)^4 - \frac{5}{6}(2M+1)^2 + 3)$
4	$8064(2M+1)^2 + 4032$
5	$64512M^2 - 5376$
6	1.154×10^7
Total	$1024M^6 - 36608M^5 + 4.4908 \times 10^5 M^4$ $+1360M^3 + 97706M^2 + 33018M + 1.1702 \times 10^7$

Table 6: Number of algebraic operations for simulating one realisation using PWSIM.

In summary, the number of algebraic operations required for simulating one conditional realisation using PWSIM is summarised in Table 6. Detail for the computation can be seen in Appendix 12.3. According to this table, for Haar wavelets 1.2248×10^7 algebraic operations have to be carried out to simulate one realisation. For Db2 wavelets the number of multiplications and additions goes up to 1.8249×10^7. However, when a set of realisations is simulated, the computation of the covariance tables is only carried out once. Therefore, suppose that 100 realisations need to be simulated the number of additions and multiplications needed are 1.1774×10^9 in the case of the Haar basis and 1.2153×10^9 in the case of the Db2 basis.

8.5.2 Computational Effort for DWSIM

The computational effort used in DWSIM includes

1. The computation of the covariance tables requires the same number of algebraic operations as in the case of PWSIM.

2. Estimation of the initial scaling image requires 2799 additions and multiplications.

3. The computation of the weights requires $3 \times (\frac{1}{3}(2M+1)^6 + \frac{33}{2}(2M+1)^4 - \frac{5}{6}(2M+1)^2 + 4)$ algebraic operations.

4. The simulation of the scaling coefficients at the finer scale via the scaling coefficient at the coarser scale requires $43008M^2 + 43008M + 16128$ additions and multiplications.

Step	N^0 of operations
1	$(5603 - 460M + 12M^2)(80M^4 - 1) + 66564$
2	2799
3	$3 \times (\frac{1}{3}(2M+1)^6 + \frac{33}{2}(2M+1)^4 - \frac{5}{6}(2M+1)^2 + 4)$
4	$43008M^2 + 43008M + 16128$
Total	$1024M^6 - 36\,608M^5 + 4.4898 \times 10^5 M^4 +$ $1168M^3 + 43831M^2 + 43751M + 79937$

Table 7: Number of algebraic operations for simulating one realisation using DWSIM.

The number of algebraic operations required for simulating one conditional realisation using DWSIM is summarised in Table 7. Detail for the computation can

be seen in Appendix 12.4. Hence, 5.4309×10^5 algebraic operations have to be performed to simulate one conditional realisation from *pH100n* in the case of the Haar basis whereas in the case of the Db2 basis the number of algebraic operations is 6.2771×10^6. As in the case of PWSIM, since the covariance tables are computed only once, if 100 realisations are to be obtained, the number of algebraic operations needed in the case of the Haar and the Db2 bases are 1.0973×10^7 and 3.3862×10^7, respectively.

8.5.3 Comparison of the Computational Effort of PWSIM and DWSIM

By comparing the computational effort for the two algorithms, one can see that DWSIM is much faster than PWSIM. Excluring the postprocessing step, PWSIM and DWSIM require approximately the same computational effort even though the methods used to obtain nonconditional realisations in PWSIM and conditional realisations in DWSIM are different. However, while in DWSIM conditional simulated realisations are obtained directly, PWSIM needs an additional postprocessing step to achieve the conditional realisations. Because of this step, PWSIM is slower than DWSIM. Based on the simulation of one realisation, in the case of the Haar wavelet basis, DWSIM is 21 times as fast as PWSIM; however, in the case of the Db2 basis DWSIM is only approximately three times as fast as PWSIM. Based on the simulation of a set of 100 realisations, DWSIM is 107 times as fast as PWSIM if the Haar basis is used and 35 times as fast as PWSIM if the Db2 basis is used.

Between the two wavelet bases, based on the simulation of one realisation, PWSIM using the Haar basis is 1.5 times as fast as in the case of the Db2 basis, and based on the simulation of 100 realisations, the computational effort using the two bases is approximately the same. On the other hand, DWSIM using the Haar basis is 11

times as fast as DWSIM using the Db2 basis if one realisation is simulated and three times as fast as DWSIM using the Db2 basis if 100 realisations are simulated.

Since the Db2 basis requires more computational effort than the Haar basis, the result obtained via DWSIM using the Db2 basis is expected to be better than that in the case where the Haar basis is used. However, contrary to the expectation and unlike the case of PWSIM, the result of the simulation using DWSIM is better in the case of the Haar basis than in the case of the Db2 basis. As can be seen from Sections 8.1 and 8.2, the goodness coefficients obtained using DWSIM with the Db2 basis in the anisotropic case are very low and the semivariogram model of the anisotropic sample B64n is not adequately reproduced.

The inappropriate reproduction of the semivariogram model can be caused by the method of obtaining the conditioning values for the coarser scales. Since the covariance among the locations of the known values in the window is not taken into account in the estimation, the larger the window size the less accurate the estimation is, especially for strongly anisotropic samples. Errors in the estimation are propagated and the compression of the study region to the initial scaling image does not strictly preserve the spatial continuity of the sample. As the consequence, even though the back simulation captures the spatial structure of the initial scaling image, this spatial structure departs from the target spatial continuity.

Since DWSIM using the Haar basis is much more efficient than DWSIM using the Db2 basis and also more efficient than PWSIM, from now on we focus only on DWSIM with the Haar basis.

9 CASE STUDY 5: THE IMPACT OF SAMPLE SIZE ON DWSIM

In this Chapter we investigate the impact of sample size on the performance of DWSIM in the case of the Haar basis. To do so, DWSIM is applied to the nested samples *pH50IRn*, *pH250IRn* and *pH500IRn* of the exhaustive data set *pHsoiln*. The results obtained by applying DWSIM using the Haar basis to *pH100IRn*, discussed in Chapters 7 and 8, are also shown for the purpose of comparison. The performance of DWSIM with regard to sample size is assessed visually and quantitatively. Visual assessment is carried out by comparing the mosaic plots of three typical realisations from each sample with the post plots of the corresponding sample and the mosaic plot of *pHsoiln*. The quantile-quantile plots of the realisations against the corresponding samples and semivariograms associated with the typical realisations are also plotted in order to visualise the reproduction of the associated sample histograms and semivariogram models. Quantitative assessment is carried out by the measures of accuracy, goodness and precision and the measures for sample histogram and semivariogram model reproduction using MAD and MAE.

9.1 Typical Realisations

The performance of DWSIM with regard to sample size is first visually assessed via mosaic plots of typical realisations, the quantile-quantile plots and experimental semivariograms. For each sample, three realisations are randomly selected from the set of 1000 realisations. In Figure 53 mosaic maps for these realisations are shown together with the post plots of the samples and the mosaic map of the corresponding exhaustive data set.

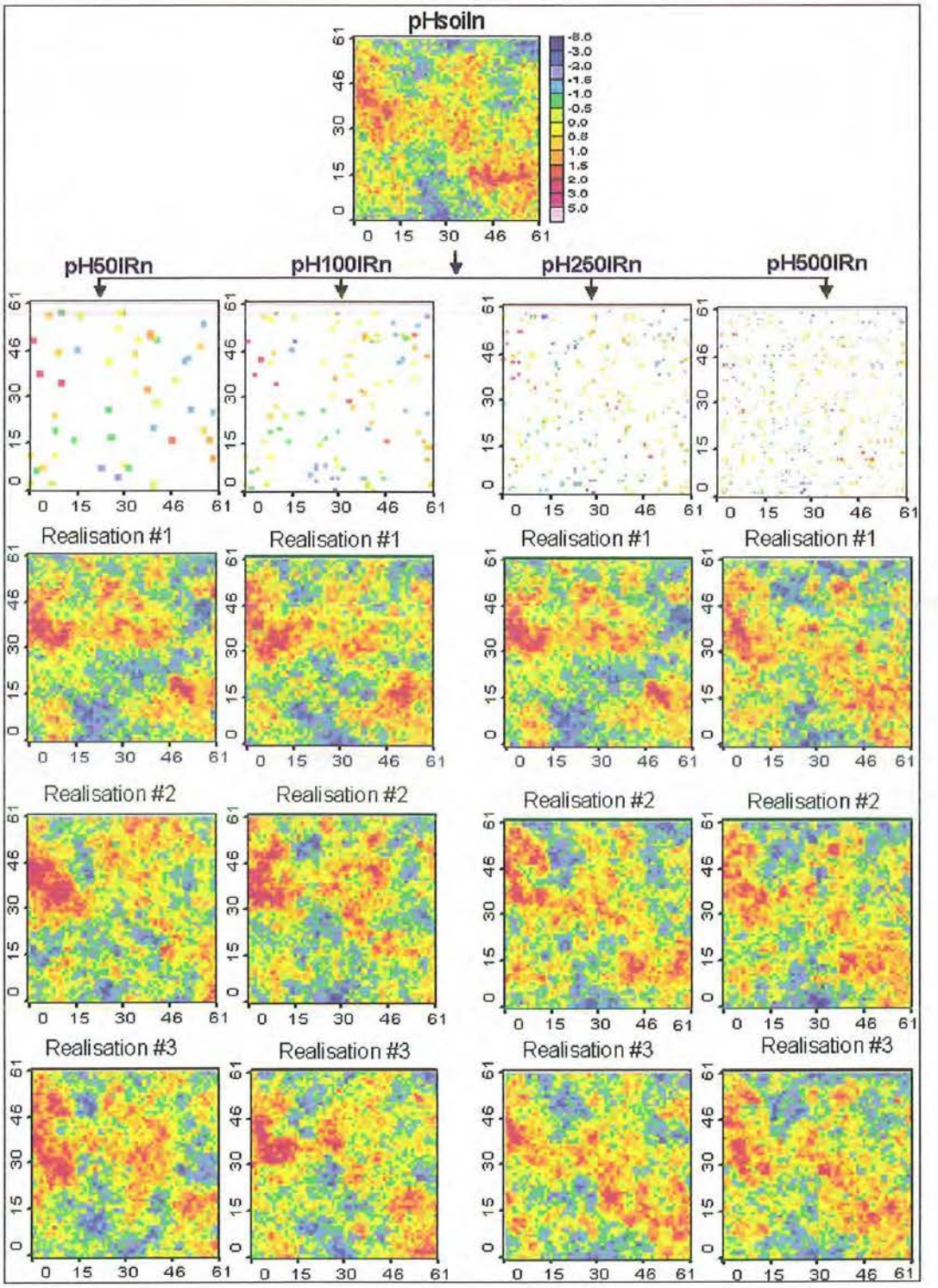


Figure 53: Mosaic plot of *pHsoiln*, post plots of nested samples and mosaic plots of three randomly selected realisations from each sample.

Summary Statistics

	pH50IRn				pH100IRn				pH250IRn				pH500IRn			
	Sample	Realisations			Sample	Realisations			Sample	Realisations			Sample	Realisations		
Stats		1	2	3		1	2	3		1	2	3		1	2	3
Mean	0.00	-0.07	0.02	0.04	-0.12	0.03	-0.02	0.01	-0.07	-0.07	0.00	0.02	0.00	0.00	-0.03	0.01
Var	1.02	1.10	1.01	1.08	1.00	1.04	1.05	1.05	1.01	1.10	1.01	1.00	1.00	1.03	1.03	1.02
Max	2.62	4.04	3.79	4.19	2.62	3.70	4.10	3.77	2.62	4.07	3.08	3.72	2.62	3.20	3.38	3.96
Q3	0.59	0.62	0.63	0.75	0.50	0.72	0.62	0.67	0.62	0.62	0.68	0.71	0.61	0.69	0.65	0.67
Med	-0.01	-0.10	-0.01	-0.01	-0.18	0.03	-0.06	-0.03	-0.01	-0.10	0.02	0.03	0.01	0.01	-0.03	0.01
Q1	-0.71	-0.76	-0.67	-0.67	-0.85	-0.65	-0.72	-0.67	-0.84	-0.76	-0.68	-0.66	-0.67	-0.67	-0.71	-0.72
Min	-1.83	-3.28	-3.19	-3.01	-2.23	-3.01	-3.50	-2.96	-2.39	-3.28	-2.97	-3.87	-2.84	-3.51	-3.72	-3.03
Skew	0.44	0.16	0.32	0.07	0.30	0.07	0.25	0.21	-0.01	0.16	0.02	0.00	0.00	-0.01	-0.02	0.13

Q-Q Plots

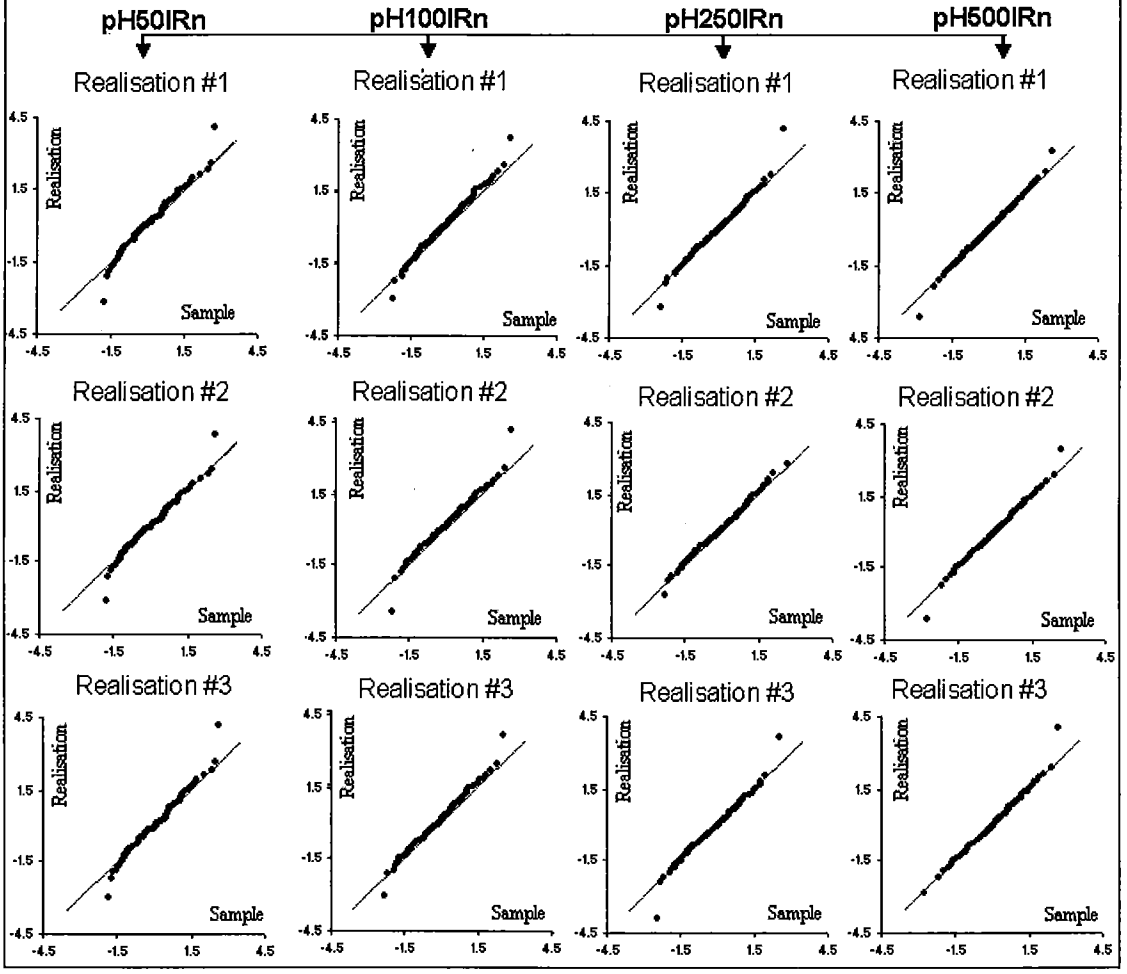


Figure 54: Summary statistics and Q-Q plots of typical realisations (from nested samples) against corresponding samples.

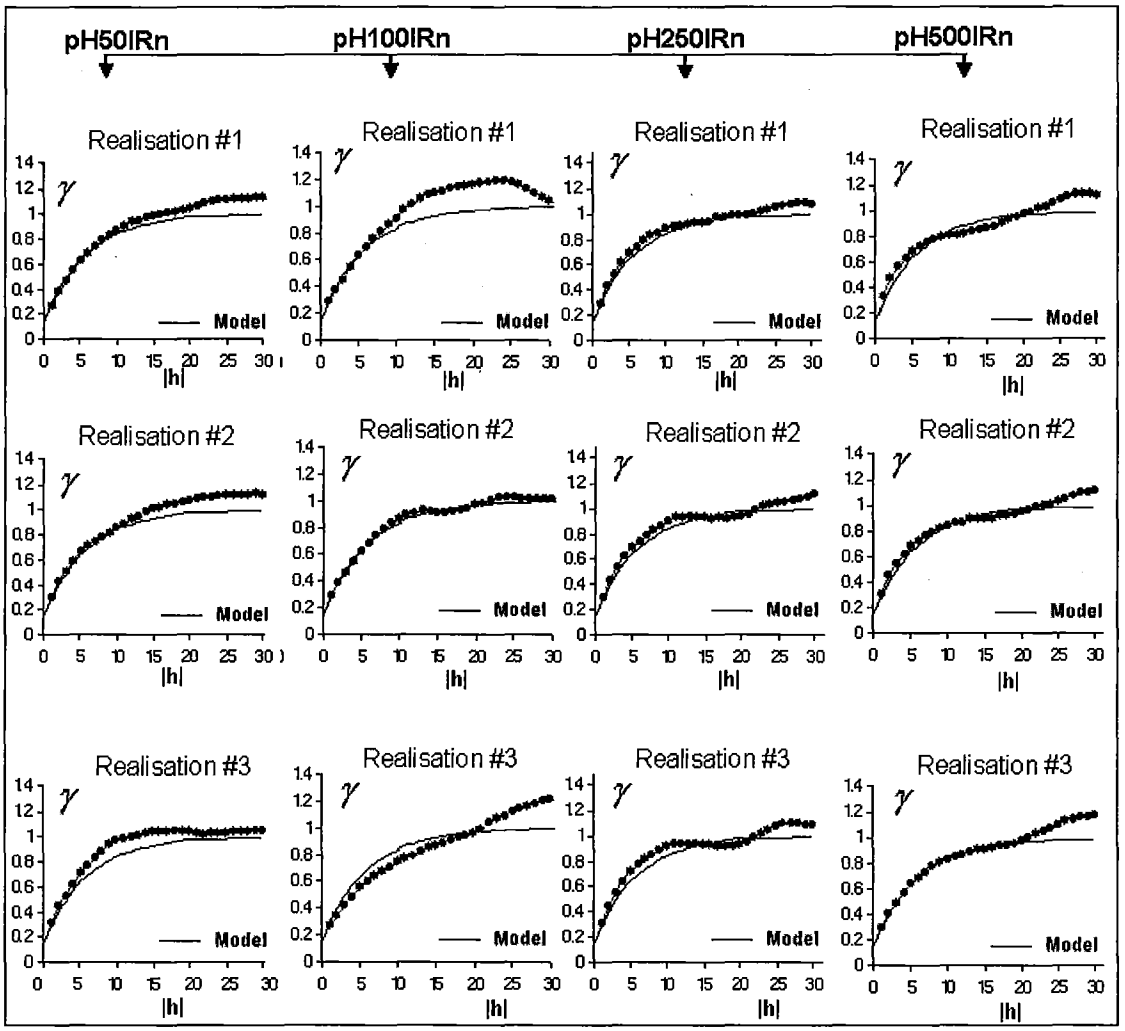


Figure 55: Experimental semivariograms of typical realisations from nested samples.

It can be seen from Figure 53 that the mosaic maps of the realisations capture all the features in the post plots of the associated samples. The more data the closer is the sample to the exhaustive data set. Therefore, as one can expect, the mosaic plots of the realisations get closer to the mosaic plot of the exhaustive data set when the sample size increases. For example, locations of high values near the left top corner and near the right bottom corner in the mosaic plot of the exhaustive data set are better reproduced in the case of realisations from *pH500IRn* than in the case of realisations from the other samples. No unusual features or artefacts are apparent in the maps.

The summary statistics together with the quantile-quantile plots of the realisations against the corresponding samples and the experimental semivariograms of the realisations in Figure 53 are shown in Figures 54 and 55, respectively. The quantile-quantile plots in Figure 54 reveal that the more data the better the sample histogram is reproduced. By comparing the summary statistics of the realisations in Figure 54 with those of the associated sample, one can see that all realisations approximately reproduce the associated sample means, variances, skewness coefficients, medians and quartiles. As a feature of the simulation, the minimum (maximum) of all realisations being lower (higher) than the corresponding sample minimum (maximum).

The experimental semivariograms in Figure 55 also indicate that the semivariogram model is reproduced regardless of sample size. All experimental semivariograms are similar to their associated semivariogram models with some fluctuation. However, the experimental semivariograms of realisations from sample *pH500IRn* show less fluctuation than those of realisations from other samples.

The reproduction of the sample histogram and semivariogram model is quantitatively assessed using the MAD and MAE values of the sets of 1000 realisations in the

following section.

9.2 Sample Histogram and Semivariogram Model Reproduction

The MAD and MAE values are used to quantitatively assess sample histogram and semivariogram model reproduction. They are calculated from 1000 realisations from each sample. Each of the MAD values is computed using 20 quantiles and each of the MAE values is computed from 18 lags using spacing of 1. The boxplots for these values are shown in Figures 56A and 56B.

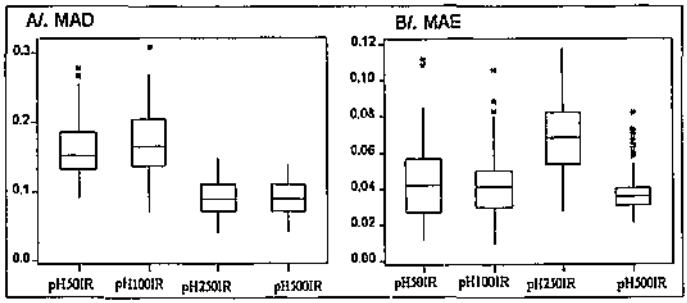


Figure 56: Box plots of the MAD and MAE values for the four samples.

In general, the sample histogram and semivariogram model of each sample are reproduced since the medians of the MAD and the MAE values are small. In the best case, the MAD and the MAE values are close to zero and in the worst case the MAD value is approximately equal to 0.3 and the MAE value is approximately equal to 0.12. For sample histogram reproduction, excluding the case of sample *pH100IR*, the boxplots in Figure 56A show that the MAD values become smaller when the sample

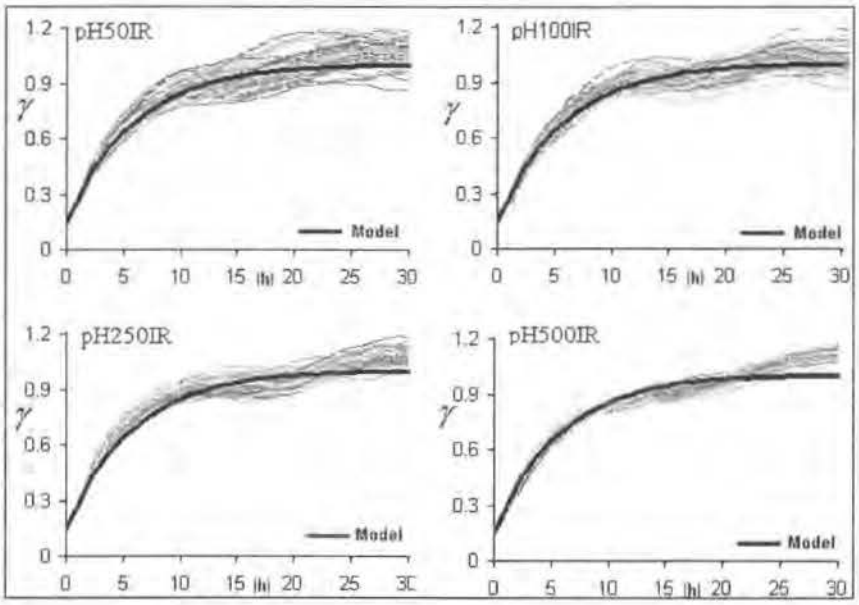


Figure 57: Experimental semivariograms of 100 realisations from the nested samples.

size increases. In other words, the more data the closer are the histograms of the realisations to the sample histogram. For semivariogram reproduction, the boxplots in Figure 56B show that, excluding the case of sample *pH250IR*, the MAE values are inversely proportional to the sample size, the more data the less deviation. The maximum error MAE decreases from approximately 0.11, in the case of *pH50IR*, to 0.10, in the case of *pH100IR*, to 0.09, in the case of *pH500IR;* and the median for the MAE also decreases from 0.04, in the case of *pH50IR*, to 0.03, in the case of *pH500IR*.

The convergence of the experimental semivariograms for the realisations of the four samples to the associated semivariogram model is illustrated via the plots in Figure 57. For readability, we only plot experimental semivariograms of the first 100 realisations from each sample. The plots show that the deviations from the model of these semivariograms decrease when the size of the sample increases. The deviations are largest in the case of *pH50IRn* and are smallest in the case of *pH500IRn*, excluding the case of *pH250IRn*.

9.3 Accuracy, Goodness and Precision of a Simulation

The increase in sample size leads to the slight improvement in the performance of a simulation assessed via the accuracy plots. Since in Figure 58 all the pairs $(p, \overline{\xi(p)})$ fall below the 45° line, only the goodness coefficients and the average variances are considered. It can be seen that while the goodness coefficient G goes from 0.96 in the case of *pH50IR*, 0.96 for *pH100IR*, 0.96 for *pH250IR* and 0.99 for *pH500IR*, the associated average variance $\overline{V}$ decreases from 0.64 to 0.56, 0.51 and 0.49, respectively. Even though the increase of the goodness coefficient G is small, the decrease of the average variance is quite considerable. Therefore, the performance of the algorithm improves with the sample size since the results are more reliable.

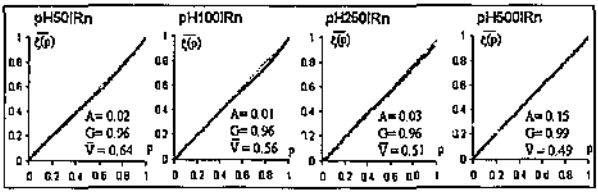


Figure 58: Accuracy plots for the nested samples.

9.4 Summary on the Impact of Sample Size

As can be seen in this Chapter, if the performance evaluation is based on the mosaic plots or the accuracy plots then increasing the sample size will increase the performance of DWSIM. If the performance evaluation is based on the reproduction of the sample histogram and semivariogram model then increasing the sample size will not strictly increase the performance of the algorithm. However, for all measures, the results obtained via DWSIM are adequate regardless of the sample size.

10 DWSIM IN COMPARISON WITH SGSIM

In this Chapter we use a practical application to compare the performance of DWSIM (using the Haar basis) with that of the sequential Gaussian simulation algorithm SGSIM (Deutsch and Journel, 1998). SGSIM is the most commonly used conditional simulation algorithm based on the assumption that the random function to be simulated follows a multivariate standard normal distribution. Because of this assumption, the isotropic sample *pHsamp* and the anisotropic sample *Bersamp* have to be transformed into normal scores before the application of the algorithms SGSIM and DWSIM.

For each sample, 1000 realisations in the normal score space are simulated. These realisations then are backtransformed to attribute values. The parameter file for SGSIM is shown in Figure 71 in Appendix 12.6.3. For the sample *pHsamp*, based on the shape of the cumulative distribution function in Figure 14, the power and hyperbolic models, each with parameter 1.5, are used for the extrapolation at the lower and upper tails, respectively. For the sample *Bersamp*, based on the shape of the cumulative distribution function in Figure 20, a linear model is used in the extrapolation at both tails. The parameter files for the backtransform are shown in Figures 72 and 73 in Appendix 12.6.4. The purpose of the extrapolation at the two tails is to allow the extreme values to lie outside the range of the corresponding sample, as the sample is only regarded as approximately representative of the population. Therefore, even though the MAD values measure the reproduction of the sample histogram, we allow the minima/maxima of the realisations to be lower/higher than the corresponding sample minimum/maximum. Since here, the exhaustive data sets are available, we set the minimum/maximum variables in the parameter file equal to the minimum/maximum of the corresponding exhaustive data set. In the case

of, *pHsamp*, the minimum and maximum pH level are set equal to 4.00 and 6.00 respectively. In the case of *Bersamp*, the minimum and maximum permeability are set equal to 19.5 and 111.5. For both isotropic and anisotropic cases, the performance comparison is based on the reproduction of the associated semivariogram models and sample histograms, the absolute errors between the average of 1000 realisations from each sample and the corresponding exhaustive data sets and the conditional variances. In addition, in the isotropic case, misclassification analysis is also carried out to compare the local accuracy of the simulated realisations obtained by the two algorithms.

10.1 Typical Realisations

We first visualise the performance of DWSIM and SGSIM by plotting the mosaic maps of the backtransform of two selected realisations from each sample together with their semivariograms and histograms. In Chapters 7, 8 and 9 the realisations were randomly selected. However, in this Chapter to visualise the reproduction of spatial continuity based on the MAE measure, the selected realisations are those that have experimental semivariograms that best and worst fit the semivatiogram models. These experimental semivariograms are computed from the normal scores of the realisations whilst the histograms are computed in terms of the attribute values. The mosaic plots for the realisations from *pHsamp* and *Bersamp* simulated via DWSIM together with the summary statistics, the quantile-quantile plots and semivariograms are shown in columns 1 and 2 and those for realisations simulated via SGSIM in columns 3 and 4 of Figures 59 and 60, respectively.

The mosaic plots, the summary statistics, quantile-quantile plots and semivariograms in Figures 59 and 60 show that both DWSIM and SGSIM maintain the

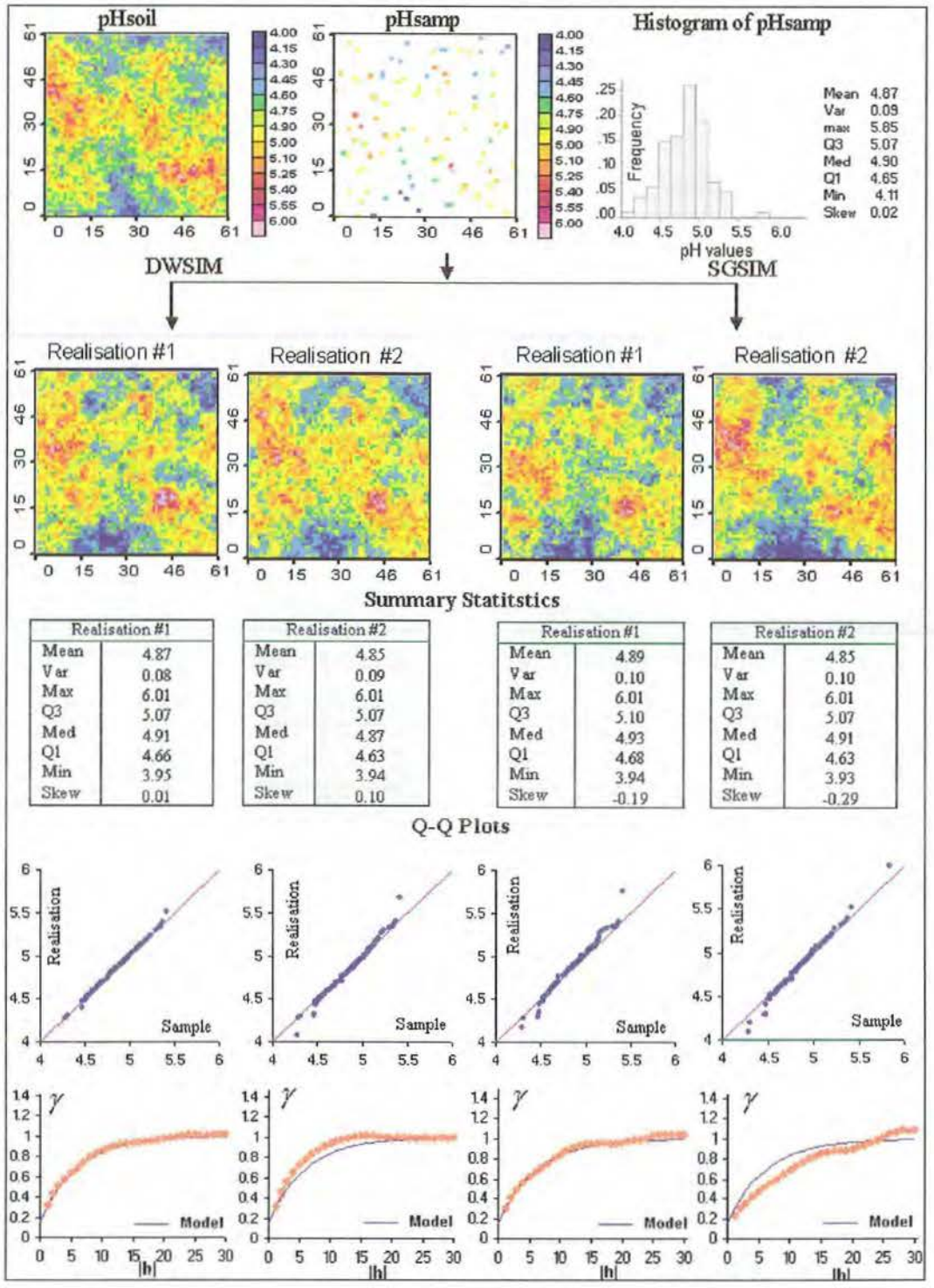


Figure 59: Mosaic plots of realisations from *pHsamp* simulated via DWSIM and SGSIM (whose semivariograms best and worst fit the model) together with Q-Q plots of the realisations against corresponding samples (using pH values) and semivariograms for normal scores.

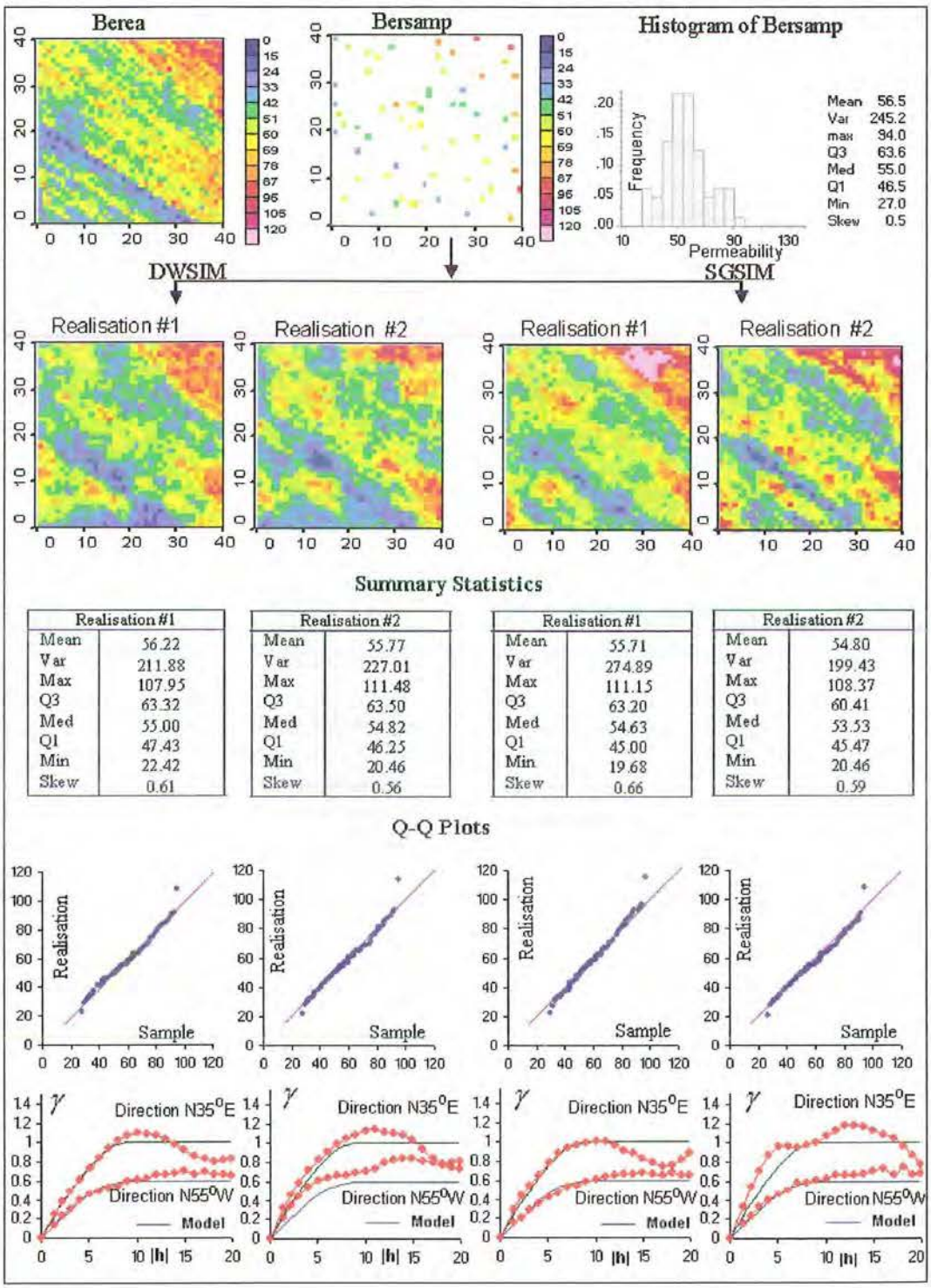


Figure 60: Mosaic plots of realisations from *Bersamp* simulated via DWSIM and SGSIM (whose semivariograms best and worst fit the model) together with Q-Q plots of the realisations against corresponding samples (using permeability values) and semivariograms for normal scores.

features, histograms and semivariogram models of the associated samples. Locations that have low and high values in the corresponding exhaustive data sets and samples are captured by the realisations. In both algorithms, the minimum/maximum of the realisations are lower/higher the corresponding sample minimum/maximum, and this is a direct consequence of the choice of the extrapolation parameters in the backtransform. In the case of sample *Bersamp*, the reproduction of the anisotropy is seen both via the banding in the mosaic maps of the realisations and from the experimental semivariograms.

10.2 Sample Histogram and Semivariogram Model Reproduction

Similar to the previous case studies, the reproduction of the sample histograms and semivariogram models is quatitatively assessed via the MAD and MAE values. These are computed from the normal scores of 1000 realisations simulated from each sample via DWSIM and SGSIM. The MAD values are computed using 20 quantiles and the MAE values are computed from 18 lags in the case of the isotropic sample and 9 lags in the case of the anisotropic sample, with lag spacing 1. Summary statistics for these values are shown by the boxplots in Figure 61 with those for the SGSIM realisations being shaded.

These boxplots reveal that, on average, in both the isotropic and the anisotropic case the sample histogram and the semivariogram model are better reproduced with DWSIM. It can be seen that the medians in the case of DWSIM are lower than those in the case of SGSIM. With regard to spatial structure of the data, DWSIM performs better in the isotropic case than in the anisotropic case whereas SGSIM performs equally well for both cases. In the case of SGSIM, the medians of the MAE values

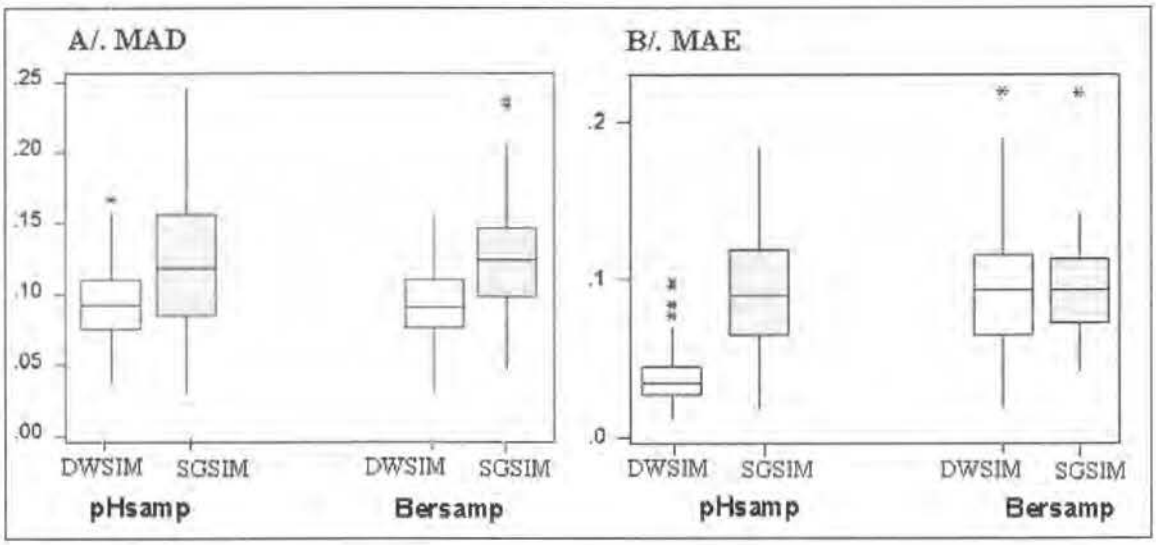


Figure 61: Boxplots of MAD and MAE values of 1000 realisations simulated via SGSIM and DWSIM.

are approximately equal for both isotropic and anisotropic samples whilst in the case of DWSIM the median of the MAE values for the isotropic sample is much lower compared with that for the anisotropic sample. With regard to the variability of the deviations, in terms of histogram reproduction (the MAD values), the variability of the deviations in the case of DWSIM is smaller than that in the case of SGSIM for both samples. In terms of semivariogram model reproduction (the MAE values), the variability in the isotropic case for realisations simulated via DWSIM is lower than the variability among realisations simulated via SGSIM. The opposite applies for the anisotropic case.

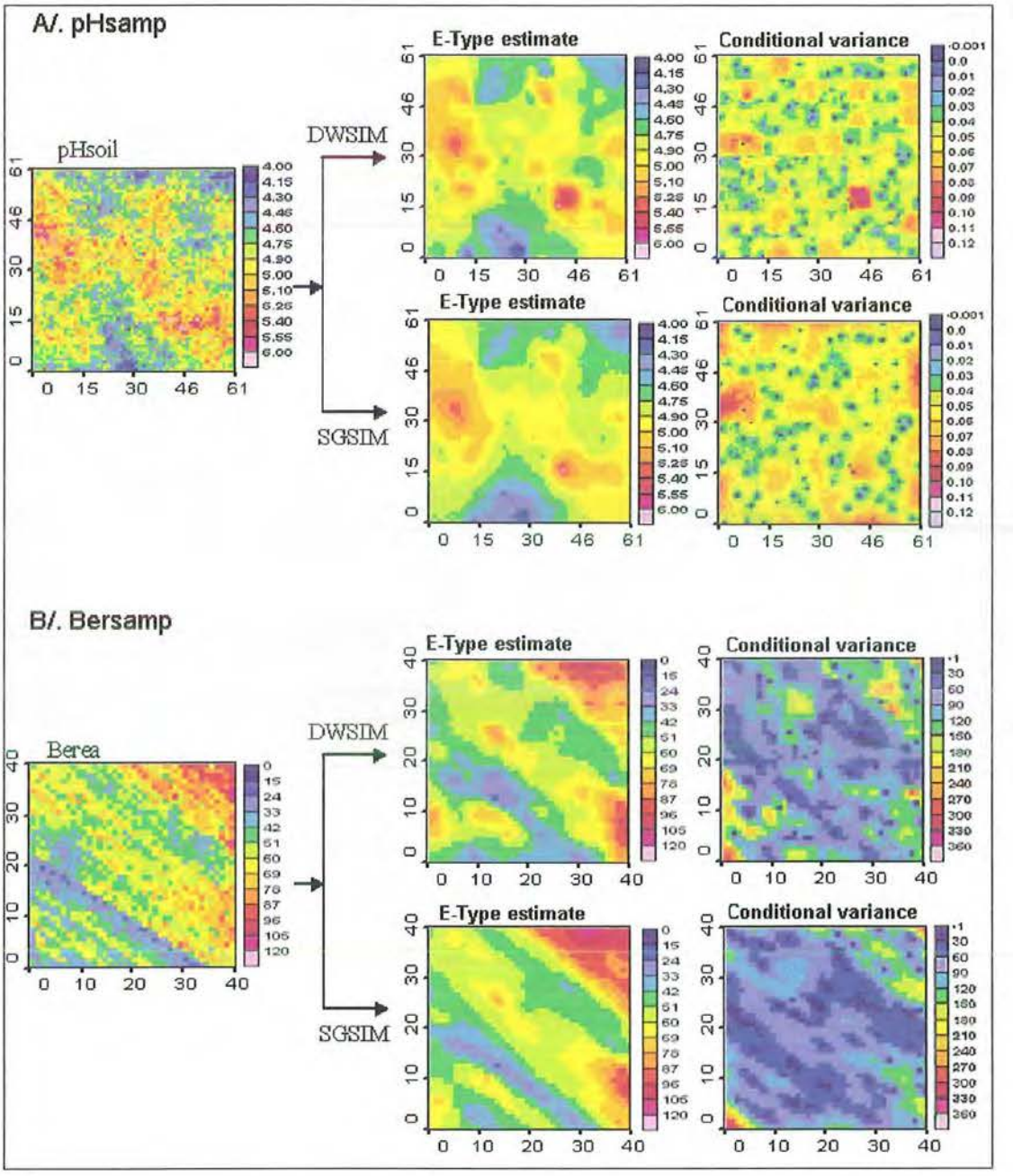


Figure 62: Mosaic maps of E-type estimates and conditional variances obtained from 1000 realisations simulated via DWSIM and SGSIM.

10.3 E-Type Estimates - Absolute Errors and Conditional Variances

E-type estimates of the realisations from *pHsamp* and *Bersamp* are obtained by averaging the backtransform of 1000 realisations simulated using DWSIM and SGSIM. The results are then compared with the values of the corresponding exhaustive data sets *pHsoil* and *Berea*. The mosaic plots of the exhaustive data sets, the E-type estimates and the conditional variances are shown in Figure 62. In the isotropic case, the mosaic maps in the middle column show that the E-type estimates from the two algorithms equally capture the features of the corresponding data set. In the anisotropic case the E-type estimate map obtained by realisations simulated via SGSIM looks closer to the mosaic map of the exhaustive data set *Berea*. The banding in the direction $N55^0W$ in the mosaic map in row 4, column 2 looks clearer compared with that in the mosaic map in row 3, column 2. The conditional variance maps in column 3 reveal that in the isotropic case the result obtained via DWSIM is more precise than that obtained via SGSIM since the local conditional variances are lower. The opposite applies for the anisotropic case. One can see from the mosaic map in row 4, column 3 that the conditional variances at locations associated with the banding and at the top right corner are lower compared with those at the same locations in the mosaic map in row 3, column 3.

The boxplots for the errors obtained by taking the absolute values of the differences between the estimates shown in Figure 62 and the true values of the corresponding exhaustive data sets *pHsoil* and *Berea* are shown in Figure 63. On average, for the isotropic sample, the median of the errors in the case of DWSIM is slightly lower than that in the case of SGSIM. The opposite is true for the anisotropic sample. In

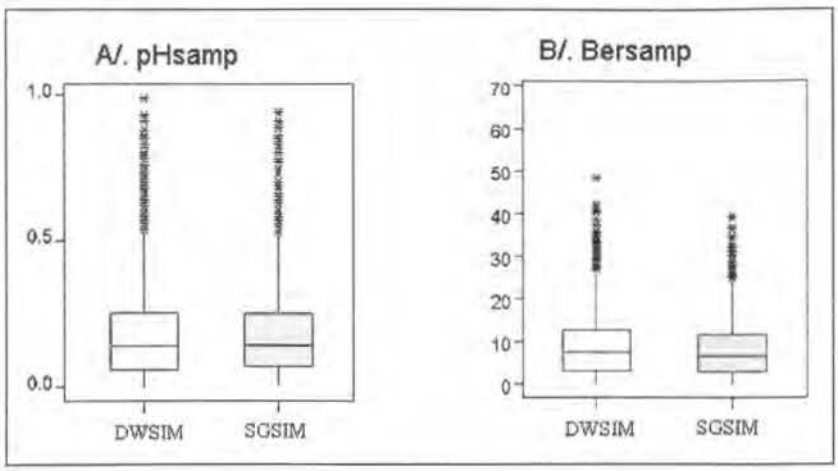


Figure 63: Boxplots for absolute of errors computed from the E-type estimates by DWSIM and SGSIM.

both isotropic and anisotropic cases, compared with the maximum errors obtained via realisations simulated via SGSIM, the maximum errors in the case of DWSIM are higher.

10.4 Misclassification Analysis

Now we compare the performance of the two algorithms in the isotropic case using misclassification analysis. In practice, for example in barley cropping, the acidity in soil is critical if the pH value is lower than 4.5. Soil acidity can lead to poor root growth which then causes poor crop yield because of nutrient deficiencies. Therefore, if the pH value drops below 4.5, some treatment, for example the application of lime to soil, needs to be carried out to reduce the acidity level. For this reason, we have taken 4.5 as the threshold for the classification. The regions where the pH values are lower than the threshold are classified as *highly acidic* whereas the locations where the pH values are higher than the threshold are classified as *moderately acidic*. An

indicator function used to classify the study region is defined as follows:

$$I(\mathbf{u}) = \begin{cases} 1 & if \quad z(\mathbf{u}) < 4.5 \\ 0 & if \quad z(\mathbf{u}) \geq 4.5 \end{cases} \tag{152}$$

Misclassification analysis is carried out for the pH values. Because of the limitation of computer space only 500 realisations are used in this performance assessment. Using the indicator function in (152), the data set *pHsoil* and the backtransformation of each of 500 realisations from sample *pHsamp*, generated by DWSIM and PWSIM, were classified into *moderately acidic* and *highly acidic* categories. The summary statistics for the percentage of the misclassification at unsampled locations, obtained by comparing the true categories with those obtained from the realisations, are shown by means of the boxplots in Figure 64, in which the boxplots for the misclassification obtained from realisations simulated via SGSIM are shaded.

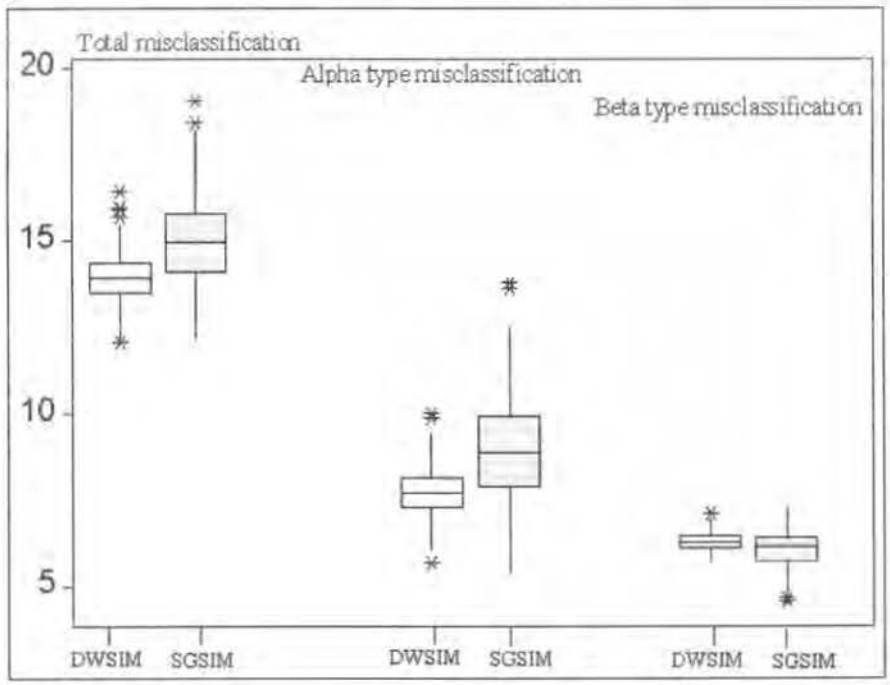


Figure 64: Boxplots for percentage misclassified locations obtained via SGSIM and DWSIM.

Based on misclassification analysis, DWSIM performs better than SGSIM since the boxplots in Figure 64 reveal that the classification using realisations simulated via DWSIM results in lower percentage of misclassified locations than the classification using realisations simulated via SGSIM. In total, the boxplots on the left of this figure show that all of the summary statistics for the percentage of misclassified locations obtained by realisations simulated via SGSIM are higher than those in the case of DWSIM. For alpha type errors (*moderately acidic* misclassified as *highly acidic*), on average DWSIM performs better than SGSIM. For beta type errors (*highly acidic* misclassified as *moderately acidic*), on average SGSIM performs slightly better than DWSIM.

The mosaic maps of the misclassified locations in Figure 65 reveal that locations that are prone to be misclassified are those that have low pH values. The mosaic plots for the total misclassification and the alpha type misclassification show that SGSIM produces more misclassified locations than DWSIM. For example, the region near the top right corner of the mosaic plots in the case of SGSIM are larger than the corresponding regions in the mosaic plots in the case of DWSIM. The differences between the mosaic maps for the beta type misclassification are negligible.

In summary, realisations simulated by DWSIM using the Haar basis and SGSIM capture the features of the associated samples and exhaustive data sets. The result of the comparison based on histogram and semivariogram model reproduction shows that the performance of the two algorithms is equivalent. Each algorithm reproduces the histograms and spatial continuity of the corresponding samples. The result of the comparison based on misclassification analysis shows that DWSIM performs slightly better than SGSIM. Finally, performance comparison based on E-type estimates and conditional variances shows that DWSIM performance and SGSIM performance are

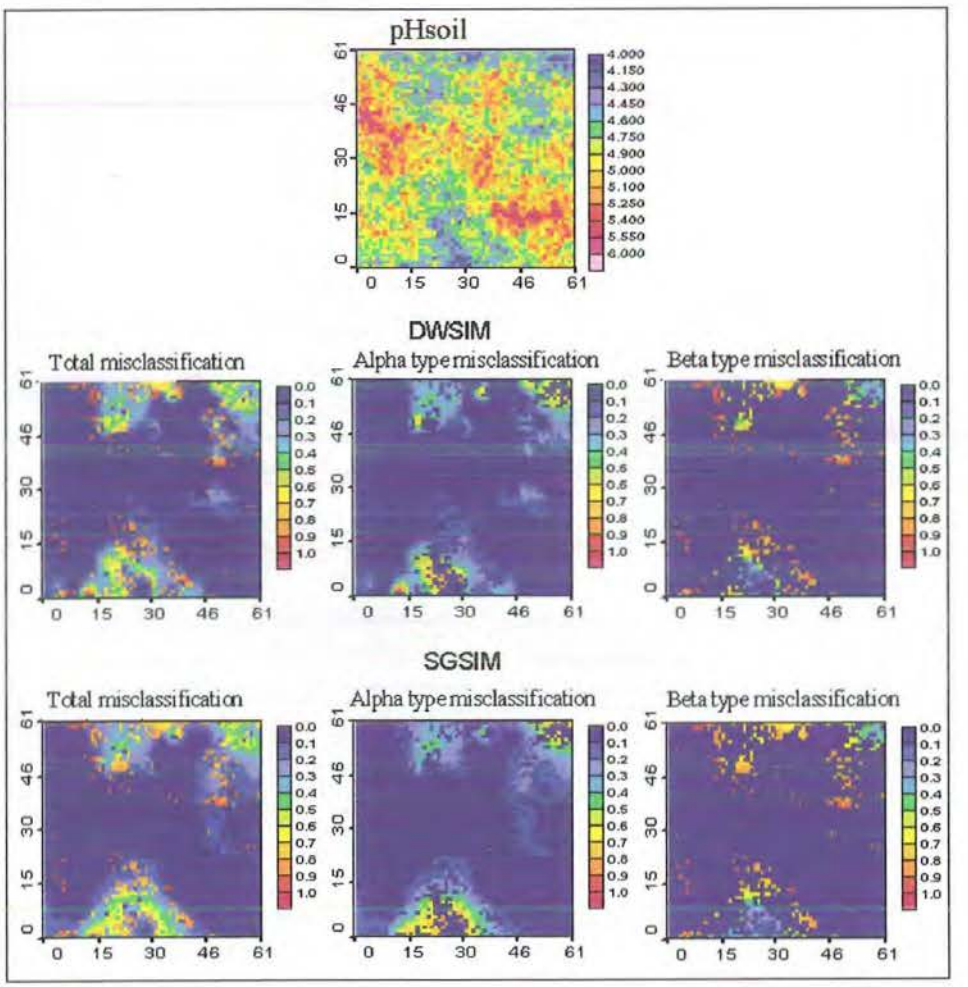


Figure 65: Mosaic maps of *pHsoil* and of misclassified locations obtained via DWSIM and SGSIM.

equivalent in the isotropic case and SGSIM performs slightly better than DWSIM in the anisotropic case.

11 CONCLUDING DISCUSSION

In this Chapter we discuss the efficiency of PWSIM and DWSIM, the impact of the wavelet basis used in the two algorithms and the result of the performance comparison between DWSIM and PWSIM and between DWSIM using the Haar basis and SGSIM.

The results obtained from the assessment of PWSIM in the case where the underlying random function is standard multivariate Gaussian in Chapters 7 and 8 show that PWSIM performs well with both wavelet bases. In general, the investigation of the mosaic maps of the simulations does not reveal any particular feature that can distinguish the differences between the mosaic plots for individual realisations simulated via the Haar basis from those simulated via the Db2 basis. All of the main features seen in the mosaic plots of the associated exhaustive data sets and samples are found in the mosaic plots of the realisations regardless of the sampling method and the spatial structure of the data. In the case of the Haar basis, even though PWSIM can start the simulation at different initial scales, as illustrated in Chapter 8, the choice of the initial scaling image size does not have a major impact on the results.

In terms of the measures of accuracy, goodness and precision, in the isotropic case where the accuracy coefficients are larger than 50%, the goodness coefficients are very close to the maximum value. In the anisotropic case where the accuracy coefficients are close to zero, the goodness coefficients are greater than 0.90. Between the two bases, the performance of PWSIM is better in the case of Db2 wavelets since the accuracy, goodness coefficients are higher, however, with the drawback that the results of the simulation are less precise.

For global accuracy, the sample histograms and semivariogram models are reproduced for each wavelet basis in both isotropic and anisotropic cases. On average, the

deviations between the histograms and the experimental semivariograms of the realisations with the associated sample histograms and semivariogram models are small. Between the two bases, this fluctuation is higher in the case of the Db2 basis. For a given wavelet basis, the fluctuation about the semivariogram model is generally higher in the case of the ungridded sample.

Based on the computational effort for one realisation, PWSIM using the Haar basis is faster; however, based on the computational effort for a set of 100 realisations the difference in the speed of the simulation is not crucial. Since the computational effort of PWSIM using the two bases is approximately similar if a set of many realisations is simulated as indicated in Chapters 7 and 8, PWSIM using the Db2 basis should be preferable to PWSIM using the Haar basis.

However, because of the postprocessing step, if compared with other conditional simulation algorithms, for example the sequential Gaussian simulation algorithm SGSIM, PWSIM requires more computation. Yet it is as complicated as SGSIM in terms of user defined parameters. In using kriging to condition the data in PWSIM, the user must determine the appropriate kriging method (ordinary or simple kriging), the search radius and the number of data to be used in the estimation.

DWSIM, on the other hand, is free from user defined parameters and is faster than PWSIM. Since DWSIM makes use of the discrete wavelet transform and its properties to obtain the conditional realisations, only the filter of the wavelet basis impacts on the result. Therefore, for a given wavelet basis the only parameters that need to be defined are those of the semivariogram model. In addition, as discussed in Chapter 8, because of the conditioning method, DWSIM is faster than PWSIM regardless of the wavelet basis used.

In the case where the underlying random function is multivariate normal, the

performance evaluation in Chapters 7, 8 and 9 shows that DWSIM using the Haar basis, is a very effective conditional simulation algorithm. The mosaic plots of the realisations simulated via DWSIM using Haar wavelets capture all relevant features of the associated exhaustive data sets and samples in both the isotropic and anisotropic cases. The sample histogram and the semivariogram model are reproduced for all cases. The goodness coefficients are close to 1 (the maximum value). The goodness coefficient is used to deduce the location of the actual outcome obtained by applying a transfer function to the true values compared with the median of the probability distribution modelled from the outcomes obtained by applying the same transfer function to a set of simulated realisations. The closer the value of the goodness coefficient is to 1, the closer to the median is the actual outcome. Since the goodness coefficient is close to 1, the predictions based on realisations simulated via DWSIM using the Haar basis are reliable.

The performance of DWSIM using the Haar basis is approximately equal to the performance of PWSIM. In terms of accuracy, goodness and precision the results obtained via DWSIM using the Haar basis are slightly more precise than those in the case of PWSIM. A drawback is that the accuracy and goodness coefficients in the case of DWSIM are slightly lower than those in the case of PWSIM. In terms of histogram and semivariogram reproduction, DWSIM using the Haar basis performs better than PWSIM. Because the performances of the two algorithms are approximately equal yet DWSIM is faster and is free from user-defined parameters, it is more efficient than PWSIM.

In this thesis, the samples we use for the performance assessment in the case where the underlying random function is not multivariate normal in both isotropic and anisotropic cases only slightly depart from the normal distibution. The perfor-

mance assessment in the case of anisotropic samples from highly skewed data sets in Tran et al, 2002a, shows that DWSIM using the Haar basis has reproduced the associated sample histogram and semivariogram models for both gridded and ungridded samples. The investigation of the mosaic maps of the realisations also reveals that the realisation ... we reproduced the spatial structure of the associated samples. No artefacts have been found in the realisations.

The performance comparison between DWSIM and PWSIM using the Haar basis in the case of a highly skewed isotropic sample in Tran et al, 2002b, indicates that realisations obtained using DWSIM have less variability than those obtained using PWSIM. In terms of sample histogram and spatial continuity reproduction DWSIM performance and PWSIM performance are approximately equivalent. In terms of misclassification analysis, the performance assessment was carried out by comparing the accuracy of realisations associated with the minimum, median and maximum MAE. In order to do so, the study region is divided into 2×2 blocks. The realisations that have minimum, median and minimum MAE obtained from PWSIM and DWSIM are back-transformed (using the same parameters) and are classified into ore or waste blocks using three cut-offs associated with the 25^{th} , 50^{th} and 75^{th} percentiles of the exhaustive data set. The category of each block then is compared with the true category obtained by classifying the exhaustive data set using the same cut-offs. The results of the comparison show that in the case of the realisation that has minimum MAE DWSIM is more accurate than PWSIM: it has a smaller number of misclassified blocks. For other realisations, PWSIM is more accurate than DWSIM.

However, if the Db2 basis is used, DWSIM is less effective especially in the anisotropic case. If the Db2 basis is used, DWSIM requires more computational effort, but the result of the simulation is no better than the result obtained if the Haar basis

is used. The semivariogram model of the gridded sample in the anisotropic case is not adequately reproduced and the goodness coefficient is low. If the goodness coefficient is too low, the actual value may fall out off the range of the outcomes obtained by the simulation. In other words, the prediction based on the simulated realisations is not reliable.

The unexpected poorer performance of DWSIM using the Db2 basis in the anisotropic case is due to the method used to estimate the initial scaling image. In the estimation stage, the covariance of the scaling coefficients within the window in the finer scale is ignored. As discussed in Chapter 8, because the number of nonzero coefficients of the Db2 filter is larger than that in the case of the Haar filter, the window in the case of the Db2 basis is larger than that in the case of the Haar basis. Hence the estimation of the scaling coefficients at the coarser scale in the case of the Db2 basis is not as appropriate as in the case of the Haar basis, especially in the anisotropic case. Therefore, in the anisotropic case the Haar basis is more appropriate for DWSIM.

In the case when the underlying random function does not follow a multivariate normal distribution, the performance comparison between DWSIM (using the Haar basis) and SGSIM in Chapter 10 shows that DWSIM performs slightly better than SGSIM in the isotropic case but the opposite is true in the anisotropic case. In the isotropic case, the histograms and experimental semivariograms of the realisations simulated via DWSIM are closer to the associated sample histograms and semivariogram models than when the realisations are simulated via SGSIM. For local accuracy, DWSIM performance and SGSIM performance are equivalent if the E-type estimates and conditional variances are compared and DWSIM performance is slightly better than SGSIM performance if the results obtained by misclassification analysis are compared. As discussed in Chapter 10, the median of the absolute errors and the con-

ditional variances computed from the realisations simulated using the two algorithms are approximately the same; but the percentage of misclassified locations in the case on SGSIM is higher than in the case of DWSIM. In the anisotropic case, the performance of the two algorithms is equivalent in terms of histogram and semivariogram reproduction. In terms of E-type estimates and conditional variances, SGSIM performance is better than DWSIM performance since the absolute errors and conditional variances are lower in the case of SGSIM than in the case of DWSIM.

In terms of computational effort, compared with SGSIM, DWSIM using the Haar basis is more efficient. Firstly, DWSIM is free from user defined parameters such as the kriging method and the number of data used in the kriging. Secondly, DWSIM does not require that the size of the study region be larger than the range of the semivariogram model. It only requires that the size of the study region be a power of 2. Thirdly, DWSIM is computationally efficient. As discussed in the end of Chapter 8, DWSIM only requires approximately 5.8209×10^5 algebraic operations to simulate one realisation from *pH100n*. For SGSIM, the number of algebraic operations used to solve for the kriging weights alone is 1.154×10^7, which is approximately 20 times the number of algebraic operations used in the case of DWSIM. If one takes into account the number of additions and multiplications used to compute the covariance among the conditioning data in the kriging system and those used to compute the error variance of the kriging estimates, the difference between the number of algebraic operations used in SGSIM and that used in DWSIM would be larger. In addition, in simulation methods usually a large number of realisations are simulated, rather than just a single realisation. If 100 realisations are required, DWSIM uses approximately only 1.0971×10^7 additions and multiplications while for SGSIM more than 1.154×10^9 algebraic operations are needed, which is 105 times as many as the number of algebraic

operations in the case of DWSIM.

In the example discussed above, the size of the region to be simulated by DWSIM is 64×64 which is approximately 61×61, the size of the study region. When the dimensions of the study region are much smaller than those of the region that DWSIM has to simulate, the computational advantage for simulating one realisation obtained using DWSIM may not be as great as in the case of the example. For example, in the anisotropic case the size of the study region is of 40×40, but the region to be simulated by DWSIM must be of size 64×64, a power of 2. Therefore, the number of grid locations to be simulated by DWSIM is much larger than the actual one whereas this requirement does not apply for SGSIM. In this case, the computational advantage of DWSIM over SGSIM is based on the large number of realisations simulated.

SGSIM is the most commonly used conditional algorithm that is based on the same assumptions as those in the case of DWSIM. This algorithm can simulate, condition the data and handle the anisotropy at the same time. However, one of the disadvantages of this algorithm over other algorithms, for example the LU decomposition simulation algorithm, is the number of user-defined parameters. In defining these parameters, one needs to validate that the choice is appropriate.

The purpose of the comparison between SGSIM and DWSIM is to find out whether or not the performance of this newly-developed algorithm DWSIM, which apart from the semivariogram model is free from user-defined parameters and which is also computationally efficient, is equivalent to that of SGSIM. Based on the example in Chapter 10, it is obvious that DWSIM performance is slightly better than SGSIM performance in the isotropic case. However, in the anisotropic case SGSIM performance is slightly better than DWSIM performance.

In summary, in this thesis we have first introduced two single-level wavelet-based

conditional simulation algorithms HSIM and DB2SIM and then moved to develop two multi-level wavelet-based conditional simulation algorithms called PWSIM and DWSIM. Both multi-level algorithms recursively reconstruct the simulated realisations from an initial scaling image at a coarse scale. While in PWSIM the result of the reconstruction is a nonconditional simulation which needs an additional postprocessing step to condition the data, in DWSIM a conditional simulation is obtained directly. Because of this, DWSIM is a very fast conditional simulation algorithm.

The performance evaluation shows that DWSIM using the Haar basis performs well based on different performance measures for both isotropic and anisotropic data. The results of the simulation obtained by DWSIM using the Haar basis are as good as those obtained via PWSIM and SGSIM yet DWSIM is more efficient than PWSIM and SGSIM. While both PWSIM and DWSIM rely on kriging, DWSIM uses the discrete wavelet transform to condition the data. This is the reason why, unlike PWSIM and SGSIM, DWSIM is fast and free from user-defined parameters. Because of these advantages, it can save a great amount of user effort in computational time and in validating the parameters.

One limitation of DWSIM is that the conditioning data have to coincide with the simulation nodes. In the case when the data do not lie on grid nodes, the application of the algorithm has to be preceded by a relocation of the data to the nearest grid nodes. In addition, DWSIM can only be applied in the case where the underlying random function is second-order stationary. For non-stationary random functions, the covariance depends on locations. The computation of the weights used in the reconstruction of the scaling image then will also depend on the locations of the simulated nodes. Hence, the method of computing the weights used in DWSIM is no longer appropriate. Therefore, more research needs to be carried out to find a

suitable approach for computing these weights so that the algorithm is still fast and effective.

12 APPENDICES

12.1 The Spectral Representation Theorem

Theorem 8 *Let $X(t)$ be a second order stationary process; $S(\omega)$ is the spectral density associated with $X(t)$ given by*

$$S(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} C(h) e^{-i\omega h} dh$$

Then there exists a unique complex stochastic process $\dot{Z}(\omega)$ satisfying (see Koopmans, 1974):

1.

$$X(t) = \int_{-\infty}^{\infty} e^{i\omega t} d\dot{Z}(\omega) \tag{153}$$

2.

$$E[d\dot{Z}(\omega)] = 0, \text{ for all } \omega$$

3.

$$E[d\dot{Z}(\omega_1)\overline{d\dot{Z}(\omega_2)}] = \begin{cases} 0 & \text{if } \omega_1 \neq \omega_2 \\ S(\omega)d\omega & \text{if } \omega_1 = \omega_2 = \omega \end{cases} \tag{154}$$

where the over-bar denotes the complex conjugate.

12.2 Checking the BiGaussian Assumption

The checking is based on the properties of a standard multivariate Gaussian random function.

Let $\mathcal{Y} : \mathcal{A} \longrightarrow \{Y(\mathbf{u}) : \mathbf{u} \in \mathcal{A}\}$ be a standard multivariate Gaussian random function. Suppose that $C_\gamma(\mathbf{h})$ is the correlogram obtained from the semivariogram model $\gamma_\gamma(\mathbf{h})$ of a sample from the random function $Y(\mathbf{u})$ by letting

$$C_\gamma(\mathbf{h}) = 1 - \gamma_\gamma(\mathbf{h}).$$

Then the two-point distribution of any pairs of the random variables $Y(\mathbf{u})$ and $Y(\mathbf{u}+\mathbf{h})$ follows normal distribution and is determined by (Goovaerts, 1997)

$$\begin{aligned} G(\mathbf{h};y_p,y_{p'}) &= Prob(Y(\mathbf{u}) \le y_p, Y(\mathbf{u}+\mathbf{h}) \le y_{p'}) && (155)\\ &= pp' + \frac{1}{2\pi}\int_0^{\arcsin C_\gamma(\mathbf{h})} \exp\left[-\frac{y_p^2 + y_{p'}^2 - 2y_p y_{p'}\sin\theta}{2\cos^2\theta}\right] d\theta && (156)\end{aligned}$$

for all $p, p' \in [0,1]$, where y_p is the standard normal p-quantile. By setting $y_p = y_{p'}$ equation (155) becomes

$$\begin{aligned} G(\mathbf{h};y_p,y_p) &= Prob(Y(\mathbf{u}) \le y_p, Y(\mathbf{u}+\mathbf{h}) \le y_p) && (157)\\ &= p^2 + \frac{1}{2\pi}\int_0^{\arcsin C_\gamma(\mathbf{h})} \exp\left[-\frac{y_p^2}{1+\sin\theta}\right] d\theta. && (158)\end{aligned}$$

The two-point probability distribution in (157) can also be written as

$$\begin{aligned} Prob(Y(\mathbf{u}) &\le y_p, Y(\mathbf{u}+\mathbf{h}) \le y_p)\\ &= E\{I(\mathbf{u},p)I(\mathbf{u}+\mathbf{h},p)\} = p - \gamma_I(\mathbf{h};p)\end{aligned}$$

where

$$I(\mathbf{u},p) = \begin{cases} 1 & \text{if } Y(\mathbf{u}) \le y_p \\ 0 & \text{if otherwise} \end{cases}$$

and $\gamma_I(\mathbf{h};p)$ is the indicator semivariogram for the threshold y_p. Therefore, the theoretical indicator semivariogram corresponding to the threshold y_p can be computed by

$$\gamma_I(\mathbf{h};p) = p - G(\mathbf{h};y_p,y_p). \qquad (159)$$

Based on equations (157) and (159), the checking for two-point Gaussian distribution is carried out by:

1. Computing the theoretical indicator semivariograms for several thresholds. This is carried out using the program BIGAUS.EXE in the GSLIB library.

2. Computing the experimental indicator semivariograms for those associated thresholds.

3. For each threshold, superimposing the experimental indicator semivariogram onto the plot of the theoretical indicator semivariogram and observing the fit. If for all thresholds, the theoretical and experimental semivariograms are similar, the biGaussian assumption is satisfied.

The checking Gaussian distribution of the samples used in this thesis is carried out by computing the theoretical and experimental indicator semivariograms for the 25-, 50- and 75-percentile of each sample. In the isotropic case, the plots of the isotropic theoretical and experimental indicator semivariograms are shown in Figure 66. In the anisotropic case, the theoretical and experimental indicator omnidirectional semivariograms are shown in Figure 67 whereas the theoretical and experimental indicator semivariograms in the directions of maximum and minimum continuity are shown in Figure 68. Based on the semivariograms, the biGaussian assumption is valid.

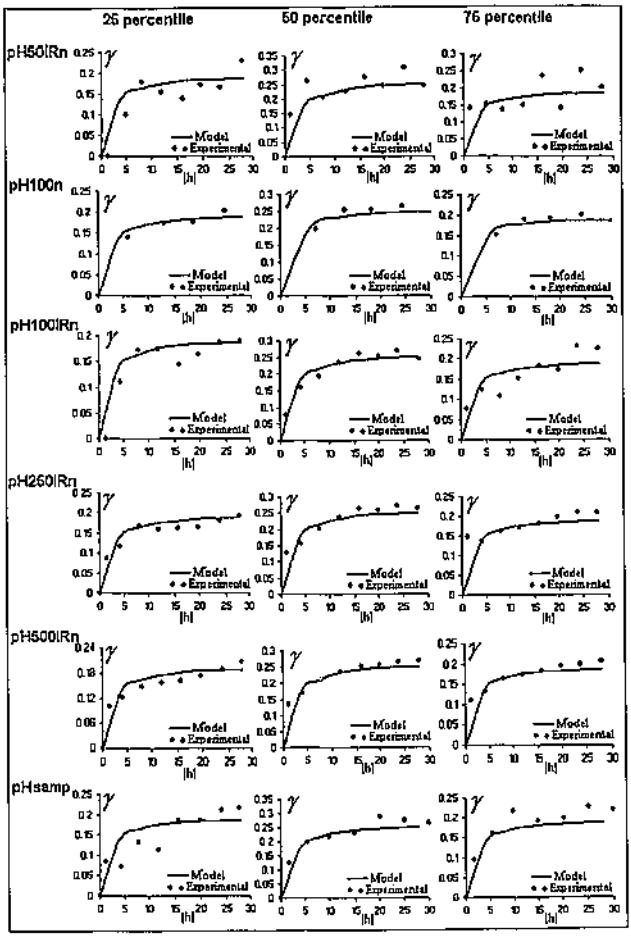


Figure 66: Theoretical and experimental indicator semivariograms for 25-, 50- and 75-percentile thresholds of isotropic samples.

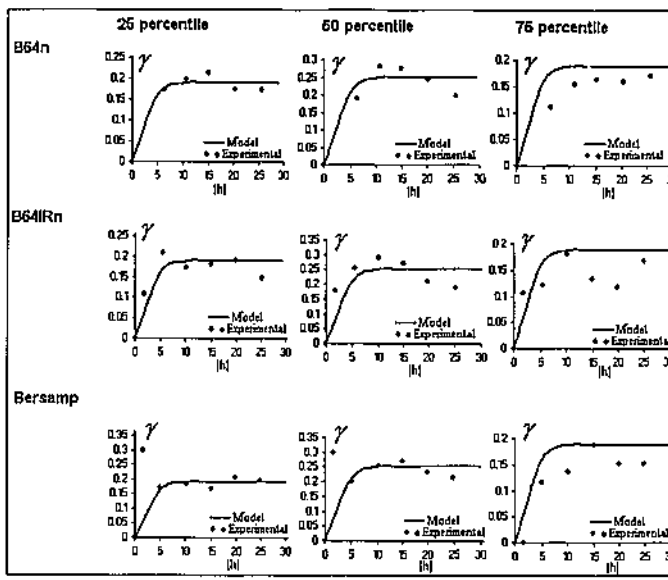


Figure 67: Theoretical and omnidirectional experimental indicator semivariograms for 25-, 50- and 75-percentile thresholds of anisotropic samples.

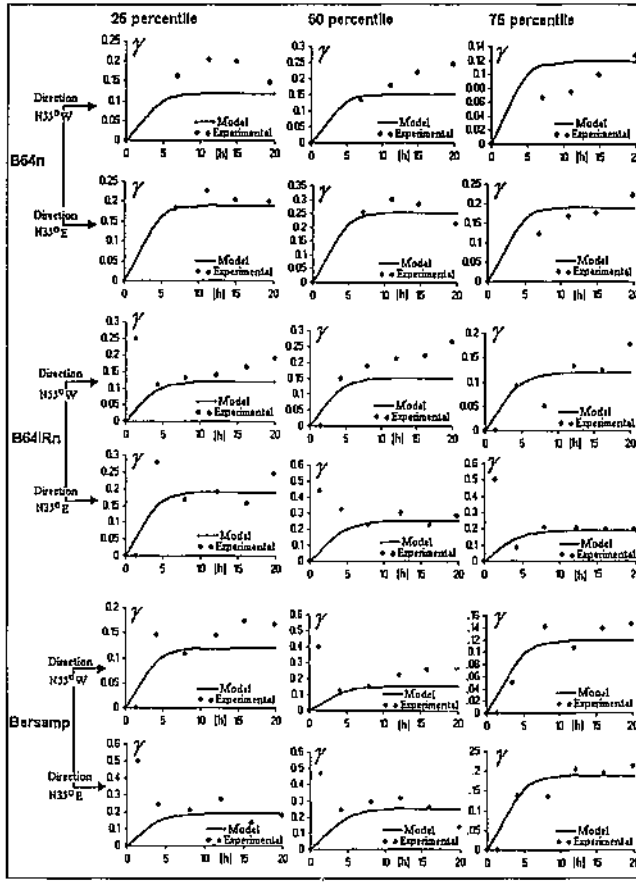


Figure 68: Theoretical and experimental indicator semivariograms for 25-, 50- and 75-percentile thresholds in directions of maximum and minimum continuity of anisotropic samples.

12.3 Computational Effort for PWSIM in Detail

1. The computation of the covariance tables: The computation of one value in the covariance table at the coarser scale from the covariance at the finer scale requires $4 \times (2M)^4$ multiplications and $(2M)^4 - 1$ additions, in total $80M^4 - 1$ operations. For a simulated realisation of size 64×64, as in our example, the number of covariance values that have to be computed in this step is shown in Table 8. Theoretically, for $j > 0$, the size of the covariance table associated with a scaling image of size $P \times Q$ is $(2P + 1) \times (2Q + 1)$. However, because of the edge effect, only $(2P + 1 - 2M) \times (2Q + 1 - 2M)$ covariance values can be computed. Hence, for this particular example, the number of algebraic operations required for the computation of the covariance tables for $j > 0$ is $(5603 - 460M + 12M^2)(80M^4 - 1)$. In addition, each value of the covariance table at $j = 0$ requires 3 multiplications and 1 addition. Therefore, in total the computation of the covariance tables for $j = 0, ..., 3$ requires $(5603 - 460M + 12M^2)(80M^4 - 1) + 66564$ operations.

j	Scaling image size	Covariance table size
0	64×64	129×129 (Not counted)
1	32×32	$(65 - 2M) \times (65 - 2M)$
2	16×16	$(33 - 2M) \times (33 - 2M)$
3	8×8	$(17 - 2M) \times (17 - 2M)$
	Total covariance values:	$5603 - 460M + 12M^2$

Table 8: The number of covariance values at j>0 to be computed.

2. The simulation of the initial scaling image: The simulation of the initial scaling image involves the Choleski decomposition of the covariance matrix and the

simulation of the associated scaling coefficients. For the covariance matrix of size $m \times m$, the decomposition requires $\frac{m^3}{6} + \frac{m^2}{2} - \frac{2m}{3}$ multiplications and $\frac{m^3}{6} - \frac{m}{6}$ additions (Burden and Faires, 1997, page 411). The simulation of associated scaling coefficients according to formula (133) needs $\frac{m(m+1)}{2}$ multiplications and $\frac{(m-1)m}{2}$ additions. In this special example, since the size of the initial scaling image is 8×8, the size of the covariance matrix is 64×64, that is $m = 64$. The decomposition of this matrix costs 89376 algebraic operations and the simulation requires 2080 multiplications and 2016 additions. In total, 93472 algebraic operations must be carried out for this step.

3. The computation of the weights: The computation of the weights in equation (134) involves:

   - The solving for the weights α_{ij} : There are three sets of weights to be solved: one for each type of wavelet coefficients. The solving for the weights is carried out using the Choleski decomposition of the coefficient matrix and the back-substitution method. The decomposition of a covariance matrix of size $m \times m$ requires $\frac{1}{3}m^3 + \frac{1}{2}m^2 - \frac{5}{6}m$ multiplications and additions. To obtain the solutions of the system of m equations, m^2 multiplications and $m^2 - m$ additions have to be carried out (Burden and Faires, 1997, page 412). Since the coefficient matrix is the same for three wavelet coefficient types, the Choleski decomposition is only carried out once. Therefore, in total, the computation of three sets of weights for the three types of wavelet coefficients involves

   $$\frac{1}{3}m^3 + \frac{1}{2}m^2 - \frac{5}{6}m + 3(2m^2 - m) = \frac{1}{3}m^3 + \frac{13}{2}m^2 - \frac{23}{6}m$$

   algebraic operations.

- The computation of the standard deviation γ of the noise: For a system of m equations, the computation of the standard deviation γ in equation 136, requires m^2+m+1 multiplications and m^2 additions. Hence, $(2m^2+m+1)\times 3$ algebraic operations are required for computing three standard deviations associated with the three sets of weights.

- Hence, to move from one coarse scale to the next finer scale, in total $\frac{1}{3}m^3+\frac{25}{2}m^2-\frac{5}{6}m+3$ algebraic operations have to be carried out. In our case, for wavelets with M vanishing moments, by replacing m with $(2M+1)^2$ on moving from a scaling image at $j=3$ to a scaling image at $j=0$, the number of algebraic operations required for computing the weights is

$$3\{\frac{1}{3}(2M+1)^6+\frac{25}{2}(2M+1)^4-\frac{5}{6}(2M+1)^2+3\}.$$

4. The simulation of the wavelet coefficients: According to formula (134), the simulation of each wavelet coefficient involves $(2M+1)^2+1$ multiplications and $(2M+1)^2$ additions. In total, $2(2M+1)^2+1$ algebraic operations have to be carried out for one wavelet coefficients. The number of additions and multiplications required to obtain all of three types of wavelet coefficients during a simulation is listed in Tables 9.

5. The inverse discrete wavelet transform: According to formula (92), the computation of each scaling coefficient at the finer scale from the scaling and wavelet coefficients at the next coarser scale via the inverse discrete wavelet transform involve $2\times(2M)^2$ multiplication and $(2M)^2-1$ additions: in total $12M^2-1$ operations. The number of algebraic operations for the inverse discrete wavelet transform is computed in Table 10.

j	Scaling image size	N^0 of operations
0	64×64	0
1	32×32	$3 \times 32 \times 32 \times (2(2M+1)^2+1)$
2	16×16	$3 \times 16 \times 16 \times (2(2M+1)^2+1)$
3	8×8	$3 \times 8 \times 8 \times (2(2M+1)^2+1)$
	Total operations:	$8064\,(2M+1)^2 + 4032$

Table 9: The number of operations used to compute all wavelet coefficients.

j	Scaling image size	N^0 Operations
0	64×64	$64 \times 64 \times (12M^2-1)$
1	32×32	$32 \times 32 \times (12M^2-1)$
2	16×16	$16 \times 16 \times (12M^2-1)$
3	8×8	0
	Total operations:	$64512M^2 - 5376$

Table 10: The number of operations used in the inverse discrete wavelet transform.

6. Conditioning the data. Assume that 16 conditioning data are used to compute the kriging estimate at one unknown location, in this example, for a study region of size 61×61 with a sample of 100 data, this step involves the solving of $3721 - 100 = 3621$ systems of 16 linear equations in 16 unknowns. Assume that the solutions are obtained by the Gaussian elimination method, according to Burden and Faires (1997) to solve a system of m equations

$$\frac{2}{3}m^3 + \frac{3}{2}m^2 + \frac{1}{2}m \tag{160}$$

algebraic operations are required. In our example, $m = 16$, therefore, 3123 algebraic operations are needed for solving one set of kriging weights. In additions, to obtain condition (33) one has to carry out $2 \times$ (16 multiplications + 15 addi-

tions) and 2 more additions (subtractions). In total, 3187 algebraic operations are required to condition one location. Therefore, in total the postprocessing step requires $3187 \times 3621 = 1.154 \times 10^7$ algebraic operations.

12.4 Computational Effort for DWSIM in Detail

1. Computation of the covariance table is the same as step 1 in Section 12.3.

2. Estimation of the initial scaling image: The number of multiplications and additions in this step depends on the number of conditioning data and their configuration. For this particular example, the sample contains 100 values, at the original scale $j = 0$ the distance between these values is 6 grid nodes. At $j = 1$, only 100 scaling coefficients are estimated from 100 windows, each of which contains only one value. The estimation of each scaling coefficient involves 3 additions (1 of which is used to increase the counter) and 2 multiplications (1 for averaging the value, 1 for doubling the average). The same number of scaling coefficients are estimated at $j = 2$ and the window associated with each estimated scaling coefficients also contains 1 value. At $j = 3$, the number of estimated values is 64 (the values of all of the grid nodes are known). Assume that each window at the finer scale associated with an estimated value contains 3 known values, the estimation of one value involves 6 additions and 2 multiplications. In addition, these values have to be rescaled so that the estimated values achieve the variance of the associated scaling coefficients. The number of additions and multiplications for the rescaling and for the estimation is listed in Table 11

3. The computation of the weights: The computation of the weights in equation

j	Estimation	Rescaling
1	300 add. + 200multi.	400 add. + 101 multi.
2	200 add. + 200multi.	400 add. + 101 multi.
3	64(7 add. + 2multi.)	256 add. + 65 multi.
4	0	0
Total	1476 operations	1323 operations

Table 11: Number of algebraic operations for the estimation stage.

(140) to move from one coarse scale to the next finer scale involves:

- The determination of four sets of weights, each for one location in the 2×2 window. Using the Choleski decomposition and the back-substitution method, the number of algebraic operations is computed in similar manner as in the case of PWSIM. However, for DWSIM, four set of weights are to be obtained. Therfore, in total the computation for four sets of weights in the case wavelets with M vanishing moments involves

$$\frac{1}{3}(2M+1)^6 + \frac{33}{2}(2M+1)^4 - \frac{5}{6}(2M+1)^2 + 4 \qquad (161)$$

algebraic operations. Therefore, on moving from a scaling image at $j = 3$ to a scaling image at $j = 0$, three times of that amount is required.

4. The simulation of the scaling coefficients at the finer scale via the scaling coefficient at the coarser scale: According to formula (140), the simulation of each scaling coefficient involves $(2M+1)^2 + 1$ multiplications and $(2M+1)^2$ additions. In total, $2(2M+1)^2 + 1$ algebraic operations have to be carried out. The total number of algebraic operations used to compute the scaling coefficients for simulating one realisation is listed in Table 12.

j	Scaling image size	N^0 Operations
0	64×64	$64 \times 64 \times (2(2M+1)^2+1)$
1	32×32	$32 \times 32 \times (2(2M+1)^2+1)$
2	16×16	$16 \times 16 \times (2(2M+1)^2+1)$
3	8×8	0
	Total operations:	$43008M^2 + 43008M + 16128$

Table 12: The number of operations used to compute the scaling coefficients in the back simulation.

12.5 Computer Programs - Data Sets and Samples

Two executive computer programs DWSIM.EXE, WSIM.EXE and the parameter files together with the data sets and samples discussed in this thesis are included on the accompanying CD. The two executive computer programs are compiled from two sets of FORTRAN 4.0 source codes. The set of source codes (not included in the CD) used to compile DWSIM.EXE consists of

- The main program *DWSIM.f*
- The include file *DWSIMInc.f*
- Subroutine *Acorni.f* from the GSLIB library used to generate uniformly distributed random numbers.
- Subroutine *Backsim.f* used to reconstruct the scaling image from a coarser scale to the previous finer scale.
- Function *Cova2.f* from the GSLIB library used to compute the covariance table at the original scale.

- Subroutine *Gauinv.f* from the GSLIB library used to generate normally distributed random numbers with mean 0 and variance 1.

- Subroutine *Getdata.f* used to read conditioning data.

- Subroutine *GetIni_Image.f* used to estimate the conditioning values at the scales associated with $j > 0$ and the initial image.

- Subroutine *Getweights.f* used to compute the weights.

- Subroutine *Scalcov.f* used to compute the covariance tables for the scales associated with $j > 0$.

- Subroutine *Readparam.f* used to read the parameters in a parameter file called *DWSIM.PAR*.

The set of FORTRAN 4.0 source codes (not included in the CD) used to compile the program *WSIM.EXE* for the non-conditional simulation consists of

- The main program *WSIM.f*

- The include file *WSIMInc.f*

- Subroutine *Acorni.f* from the GSLIB library used to generate uniformly distributed random numbers.

- Subroutine *Backsim.f* used to reconstruct the scaling image from a coarser scale to the previous finer scale.

- Function *Cova2.f* from the GSLIB library used to compute the covariance table at the original scale.

- Subroutine *Gauinv.f* from the GSLIB library used to generate normally distributed random numbers with mean 0 and variance 1.
- Subroutine *Getweights.f* used to compute the weights for simulating the wavelet coefficients.
- Subroutine *Hscalcov.f* used to compute the covariance table for the scales associated with $j > 0$ using the Haar lowpass filter.
- Subroutine *Readparam.f* used to read the parameters in a parameter file called *WSIM.PAR*.

The data sets and samples included in the CD are

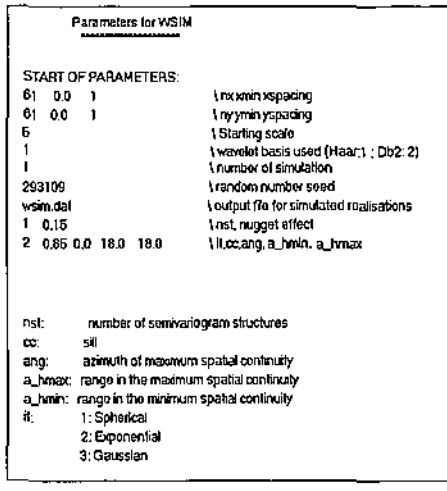
- Data set *pHsoil*.
- Data set *pHsoiln*.
- Sample *pH100n* and the jackknife data *JackpH100n*.
- Sample *pH100IRn* and the jackknife data *JackpH100IRn*.
- Sample *pH50IRn* and the jackknife data *JackpH50IRn*.
- Sample *pH250IRn* and the jackknife data *JackpH250IRn*.
- Sample *pH500IRn* and the jackknife data *JackpH500IRn*.
- Data set *Berca*.
- Data set *Berean*.
- Sample *B64n* and the jackknife data *JackB64n*.
- Sample *B64IRn* and the jackknife data *JackB64IRn*.

- Sample *pHsamp*
- Sample *Bersamp* to simulate conditional realisations using the Haar and Db2 bases. It consists of

12.6 Parameter Files

The following parameter files should be opened using WordPad.

12.6.1 Parameter File for WSIM



Parameters for WSIM

START OF PARAMETERS:
61 0.0 1 \ nx xmin xspacing
61 0.0 1 \ ny ymin yspacing
6 \ Starting scale
1 \ wavelet basis used (Haar:1 ; Db2: 2)
1 \ number of simulation
293109 \ random number seed
wsim.dat \ output file for simulated realisations
1 0.15 \ nst, nugget effect
2 0.85 0.0 18.0 18.0 \ it,cc,ang, a_hmin, a_hmax

nst: number of semivariogram structures
cc: sill
ang: azimuth of maximum spatial continuity
a_hmax: range in the maximum spatial continuity
a_hmin: range in the minimum spatial continuity
it: 1: Spherical
2: Exponential
3: Gaussian

Figure 69: Parameter file for WSIM.

12.6.2 Parameter File for DWSIM

```
                Parameters for DWSIM
                ********************

START OF PARAMETERS:
ber64n.dat                              \ data file
1   2    3    64                        \ columns for x, y, variable,numdata
40  0.5  1                              \ nx xmin xspacing
40  0.5  1                              \ ny ymin yspacing
1                                       \ wavelet basis used (Haar:1 ; Db2: 2)
1000                                    \  number of simulation
283107                                  \ random number seed
dwsim.dat                               \ output file for simulated realisations
2   0.0                                 \ nst, nugget effect
1   0.6  305.0  9.0   1000.0            \ it,cc,ang,a_hminx, a_hmax
1   0.4  35.0   9.0   9.0               \it,cc,ang,a_hminx, a_hmax

nst:      number of semivariogram structures
cc:       sill
ang:      azimuth of maximum spatial continuity
a_hmax:   range in the maximum spatial continuity
a_hmin:   range in the minimum spatial continuity
it:       1: Spherical
          2: Exponential
          3: Gaussian
```

Figure 70: Parameter file for DWSIM.

12.6.3 Parameter File for SGSIM

```
                  Parameters for SGSIM
                  ********************

START OF PARAMETERS:
pHsampn.dat                  \ file with data
1  2  0  4  0  0             \ columns for X,Y,Z,vr,wt,sec.var,
-1.0e21      1.0e21          \ trimming limits
0                            \ transform the data (0=no, 1=yes)
sgsim.trn                    \ file for output trans table
0                            \ consider ref. dist (0=no, 1=yes)
histsmth.out                 \ file with ref. dist distribution
1  2                         \ columns for vr and wt
0.0   15.0                   \ zmin,zmax(tail extrapolation)
1      0.0                   \ lower tail option, parameter
1     15.0                   \ upper tail option, parameter
1                            \ debugging level: 0,1,2,3
sgsim.dbg                    \ file for debugging output
sgsim.dat                    \ file for simulation output
1000                         \ number of realizations to generate
61    0.0    1.0             \ nx,xmn,xsiz
61    0.0    1.0             \ ny,ymn,ysiz
1     0.0    1.0             \ nz,zmn,zsiz
7929708                      \ random number seed
4     20                     \ min and max original data for sim
12                           \ number of simulated nodes to use
1                            \ assign data to nodes (0=no, 1=yes)
1     3                      \ multiple grid search (0=no, 1=yes),num
20                           \ maximum data per octant (0=not used)
20.0  20.0  10.0             \ maximum search radii (hmax,hmin,vert)
 0.0   0.0   0.0             \ angles for search ellipsoid
1     0.60   1.0             \ ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC
../data/ydata.dat            \ file with LVM, EXDR, or COLC variable
4                            \ column for secondary variable
1    .15                     \ nst, nugget effect
2    .85 0.0   0.0   .0      \ it,cc,ang1,ang2,ang3
```

Figure 71: Parameters for SGSIM.

12.6.4 GSLIB Parameter Files for Back-transformation

```
                  Parameters for BACKTR
                  *********************

START OF PARAMETERS:
dwsimxy.dat                 \ file with data
3                           \ column with Gaussian variable
-1.0e21   1.0e21            \ trimming limits
dwsimb.dat                  \ file for output
pHsampn.trn                 \ file with input transformation table
 4     6                    \ minimum and maximum data value
 2     1.5                  \ lower tail option and parameter
 4     1.5                  \ upper tail option and parameter
```

Figure 72: Parameter file for backtransformation of realisations from *pHsamp*.

```
                  Parameters for BACKTR
                  *********************

START OF PARAMETERS:
dwsim.dat                   \ file with data
1                           \ column with Gaussian variable
-1.0e21   1.0e21            \ trimming limits
dwsimb.dat                  \ file for output
bersampn.trn                \ file with input transformation table
 19.5    111.5              \ minimum and maximum data value
 1     0                    \ lower tail option and parameter
 1     0                    \ upper tail option and parameter
```

Figure 73: Parameter file for backtransformation of realisations from *Bersamp*.

12.7 Notation

$\oplus$: direct sum

$\otimes$: tensor product

$\langle f, g \rangle$: inner product

A : accuracy coefficient

$\mathcal{A}$: study region

a: range of semivariogram

$\beta^j_{(m,n)}$: covariance between two scaling coefficients $c^j[m]$ and $c^j[n]$ in one-dimensional space

$\beta^j_{(p,m),(q,n)}$: covariance between two scaling coefficients $c^j[p,q]$ and $c^j[m,n]$ in two-dimensional space

$c^j[n]$: one-dimensional scaling coefficient at location n

$c^j[m,n]$: two-dimensional scaling coefficient at location (m,n)

$C(\mathbf{h})$: covariance function of stationary random function Z for lag vector h

$C(0)$: covariance value at separation distance $|\mathbf{h}| = 0$

$C(\mathbf{u}, \mathbf{u}')$: non-stationary covariance of random variables $Z(\mathbf{u})$ and $Z(\mathbf{u}')$.

$d^j[n]$: one-dimensional wavelet coefficient at location n

$d^j_k[m,n], k = 1, 2$ or 3 : horizontal, vertical or diagonal two-dimensional wavelet coefficient at location (m,n)

$\delta^j_{(m,n)}$: covariance between two scaling coefficients $c^j[m]$ and $c^{j+1}[n]$

$\delta^j_{(p,m),(q,n)}$: covariance between two scaling coefficients $c^j[p,q]$ and $c^{j+1}[m,n]$

$E\{.\}$: expected value

$\eta^j_{(m,n)}$: covariance between wavelet coefficient $d^j[m]$ and scaling coefficient $c^j[n]$

$\eta^j_{k(m,p),(n,q)}$: covariance between wavelet coefficient $d^j_k[p,q], k = 1, 2$ or 3 and scaling coefficient $c^j[n,m]$

$\widehat{F}(\mathrm{u}; z|(n))$: conditional cumulative distribution function at location u of a random variable $Z(\mathbf{u})$ with regard to the information given by n data

$g(|\mathrm{h}|)$: isotropic standardised semivariogram model

$g[n]$: wavelet highpass filter

G : goodness coefficient.

$\gamma(\mathrm{h})$: semivariogram function of stationary random function Z for lag vector h.

$h[n]$: wavelet lowpass filter

$L^2(\mathbb{R})$: space of square integrable functions such that $\int_{-\infty}^{\infty} |f(t)|^2 dt < \infty$ in one-dimension

M : number of vanishing moments.

n : number of conditioning data in the study region A.

$n(\mathrm{u})$: number of data value used for estimation the attribute value at location u

$N(0, \sigma^2)$: normal distribution with mean 0 and variance σ^2

ϕ : one- or two-dimensional father wavelet

$\phi_{j,n}$: one-dimensional scaling function obtained by dilating and shifting the father wavelet a scale factor 2^j and a translation factor n

$\phi_{j,m,n}$: two-dimensional scaling function obtained by dilating and shifting the father wavelet a scale factor 2^j and translation factors m and n in horizontal and vertical directions

ψ : one-dimensional mother wavelet

$\psi_{j,n}$: one-dimensional wavelet obtained by dilating and shifting the mother wavelet a scale factor 2^j and a translation factor n

ψ^k : two-dimensional mother wavelet

$\psi^k_{j,m,n} (k = 1, 2 \text{ or } 3)$: horizontal, vertical or diagonal two-dimensional wavelet obtained by dilating and shifting the mother wavelet a scale factor 2^j and translation factors m and n in horizontal and vertical directions

$\rho(\mathrm{h})$: correlogram function of stationary random function Z for lag vector h

$\mathbb{R}$: set of real numbers

$\sigma^2(\mathrm{u})$: conditional variance of random variable Z at location u

$\tau^j_{(m,n)}$: covariance between two wavelet coefficients $d^j[m]$ and $d^j[n]$

$\tau^j_{k(m,p),(n,q)}, k = 1, 2 \text{ or } 3$: covariance between two wavelet coefficients $d^j_k[m, n]$ and $d^j_k[p, q]$

u : coordinate vector at a location

$\overline{V}$: average variance.

V_j : space of all approximations of the function f at the scale 2^j

W_j : detail space, orthogonal complement of V_j

ξ^j : scaling spectrum

$\mathbb{Z}$: set of integers

Z : random function

$z(\mathbf{u})$: true value at location u

$z_s(\mathbf{u})$: nonconditional simulated value at location u

$z_{sc}(\mathbf{u})$: conditional simulated value at location u

$z^*_K(\mathbf{u})$: kriging estimate of the true value $z(\mathbf{u})$ at location using data values at sample locations

$z^*_{sK}(\mathbf{u})$: kriging estimate of the true value $z(\mathbf{u})$ at location using simulated values at sample locations

$Z(\mathbf{u})$: continuous random variable at location u

$Z^*_{SK}(\mathbf{u})$: simple kriging estimator of $Z(\mathbf{u})$

$Z^*_{OK}(\mathbf{u})$: ordinary kriging estimator of $Z(\mathbf{u})$

$\zeta^j_k (k = 1, 2 \text{ or } 3)$: wavelet spectrum

References

[1] Bloom, M. L., Kentwell, D. J., (1999). A geostatistical analysis of cropped and uncropped soil in the Jimperding Brook catchment area of Western Australia. In: *Gomez-Hernandez, J. et al (Eds), GeoENV II, Geostatistics for Environmental Applications.* Kluwer Academic Publishers, Dordrecht, London, pp. 369-380.

[2] Burden, R. L. and Faires, J. D. (1997), *Numerical Analysis*, 6^{th} Ed, Brooks/Cole Publishing Company,USA, 811 pages.

[3] Chilès J-P. and Delfiner, P. (1999): *Geostatistics Modelling Spatial Uncertainity*, John Wiley & Sons, Inc., New York, pp 695.

[4] Daubechies, I. (1988). Orthonormal Bases of Compactly Supported Wavelets, *Comm. Pure Appl. Math.*, V. 41, pp. 909-996.

[5] Daubechies, I. (1992). *Ten Lecture on Wavelets*, The Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 357 pp.

[6] Davis, M. W. (1987). Production of conditional simulations via the LU triangular Decomposition of the covariance matrix, *Mathematical Geology*, V. 19, N. 2, 91-98.

[7] Deutsch, C. V. (1997). Direct Assessment of Local Accuracy and Precision in *Geostatistics Wollongong'96*, V.1, edited by Baafi. E. Y and Schofield N. A. Kluwer Academic Publishers, The Netherlands, 115-125.

[8] Deutsch, C. V. and Journel, A. G. (1998). *GSLIB: Geostatistical Software Library Users Guide*, 2^{nd} Edition. Oxford University Press, 369 pp.

[9] Gerald, C. and Wheatley, P. O. (1994). *Applied Numerical Analysis*, 5th Ed. Addison-Wesley Publishing Company, USA, 796 pp.

[10] Giordano, R., Slater, S. and Mohanty, K. (1985). The effects of permeability variations on flow in porous media. *SPE Paper 14365, 60th SPE Annual Conference*, Las Vegas.

[11] Goovaerts, P. (1997). *Geostatistics for Natural Resources Evaluation*, Oxford University Press, New York, 483 pp.

[12] Goovaerts, P. (2003). Geostatistical Assessment of the Spaces of Local, Spatial and Response Uncertainty for Continuous Petrophysical Properties, Chapter in book *Stochastic Modelling II* published by the American Association of Petroleum Geologists.

[13] Isaaks, E. H. and Srivastava, R. M. (1989). *Applied Geostatistics*, Oxford University Press, New York, 561 pp.

[14] Journel, A. G. and Huijbregts, C. J. (1978). *Mining Geostatistics*. Academic Press. London, 600 pp.

[15] Kahane, J. P and Lemarie-Rieusset, P-G., 1995. *Fourier Series and Wavelets*, Gordon and Breach, Australia, 394 pp.

[16] Koopmans, L.H. (1974).*The Spectral Analysis of Time Series*, Academic, New York, 366 pp.

[17] Mallat, S. (1998). *A Wavelet Tour of Signal Processing*. Academic Press, 577 pp.

[18] Pardo-Iguzquiza, E. and Chica-Olmo, M. (1993). The Fourier Integral Method: an Eddicient Spectral Method for Simulation of Random Fields, *Mathematical Geology*, V. 25, N. 2, 177-217.

[19] Rudin, W. (1991) *Functional Analysis*. McGraw-Hill, Inc., 424 pp.

[20] Sachs, R.V, Nason, G.P., Kroisandt, G. *(2000)*. Wavelet Processes and Adaptive Estimation of the Evolutionary Wavelet Spectrum, *Journal of the Royal Statistical Society, Series B*, V. 60, N. 2, 271-292.

[21] Tran, T., Mueller, U. and Bloom, L. (2001). Conditional Simulation of Two-Dimensional Random Processes Using Daubechies Wavelets, *MODSIM 2001, International Congress on Modelling and Simulation*, Edited by F. Ghassemi, D. White, S. Cuddy and T. Nakanishi, Modelling and Simulation Society of Australia and New Zealand Inc., V. 4, 1991-1996,

[22] Tran, T., Mueller, U. and Bloom, L. (2002a). Multi-level conditional simulation of two-dimensional random processes using Haar wavelets, *Proceedings of the GAA Symposium on Quantifying Risk and Error*, Perth, Australia, 56-78.

[23] Tran, T., Mueller, U. A. and Bloom, L. M. (2002b). Conditional simulation via Haar wavelets: A comparison, in *Bayer, U. et al (eds) IAMG 2002 Proceedings of the 8th Annual Conference of the International association for Mathematical Geology*, Terra Nostra, Berlin 3, 209-214.

[24] Vann J., Bertoli O. and Jackson S. (2001). An Overview of Geostatistical Simulation for Quantifying Risk, *Proceedings of the GAA Symposium on Quantifying Risk and Error*, Perth, Australia, 13-27.

[25] Vannuci, M. and Corradi, E. (1997). Some Findings on the Covariance Structure of Wavelet Coefficients: Theory and Models in a Bayesian Perspective. *Report UKC/IMS/97/05, Institute of Maths and Stats, University of Kent at Canterbury, UK*, 24 pages.

[26] Yao, T. (1998). Conditional Spectral Simulation with Phase Identification, *Mathematical Geology*, V. 30, N. 3, 1998, 285-308.

[27] Zeldin, B. and Spanos, P. (1995). Random Field Simulation Using Wavelet Bases. In *Applications of Statistics and Probability*, Lemaire,M., Favre J-L and Mebarki A. (eds), Balkema, Rotterdam.